

Подход к распознаванию полей гибкой формы на примере даты окончания действия кредитной карты

Шешкус Александр¹, Николаев Дмитрий², Ингачева Анастасия³,
Скорюкина Наталья⁴

¹ Smart Engines Ltd, Victoria, Seychelles
asheshkus@smartengines.biz

^{2,3} ИППИ РАН им. Харкевича, 127051, Москва, Россия
dimonstr@iitp.ru, ingacheva@iitp.ru

⁴ НИТУ «МИСиС», 119049, Москва, Россия
skleppy.inc@gmail.com

Abstract. В данной работе рассматривается задача поиска информационных полей документа с гибкой формой на примере распознавания даты окончания срока действия кредитной карты. Обсуждаются принципиальные трудности этой задачи и предлагаются методы ее решения. Рассматриваемая задача решается для случая применения на мобильных устройствах, что накладывает жесткие требования на вычислительную сложность. В работе приводятся результаты формального анализа производительности и точности предложенного алгоритма. Спектр ошибок системы распознавания как целого показывает, что предложенный алгоритм решает задачу с требуемой точностью.

Keywords: Распознавание, детекция текста, распознавание гибких форм

1 Введение

В наши дни программы, предназначенные для автоматического распознавания какой-либо информации, встречаются все чаще и чаще. Особенно актуальны подобные приложения для мобильных устройств (смартфоны, планшеты и т.п.). Сюда входит широкий спектр программ, от приложений по распознаванию улыбки на фотографии до пакетов распознавания текста документов. Важно отметить, что мобильное устройство, как правило, не оснащено высокопроизводительным аппаратным обеспечением, что накладывает определенные ограничения на вычислительную сложность применяемых алгоритмов. Кроме того, следует учесть, что распознаванием с мобильного устройства занимаются конечные пользователи (владельцы телефона), а не специально подготовленные операторы, что требует предельной автоматизации и устойчивости программ.

Распознавание данных с кредитной карты является весьма интересной задачей с точки зрения возможных алгоритмических подходов. При этом задача также и актуальна – подобная программа может избавить человека от надобности вводить большую часть данных собственной банковской карты вручную при осу-

ществлении покупок в интернете. Следует отметить, что современные пользователи уже вполне доверяют системам онлайн-платежей (доля которых на рынке неуклонно растет, что можно видеть из отчета «World Payments Report 2014» [1]). Конечно, можно сохранить данные карты и при покупках пользоваться авто заполнением, но по нормам безопасности подобные данные не следует хранить на мобильном устройстве, так как устройство может потеряться или быть украдено. Программа распознавания пластиковых карт поможет избежать хранения секретных данных на устройстве, с одной стороны, и позволит не вводить их каждый раз заново, с другой. Для удобства пользования подобной системой качество ее работы должно быть высоко, а время распознавания минимально.

Надо сказать, что к указанной проблеме подходят с разных сторон. Например, компанией «Google» запатентована концептуальная схема решения задачи распознавания кредитной карты, причем она подразумевает предварительную оценку качества изображения на устройстве, а распознавание – на сервере [2]. Недостаток такого подхода в том, что скорость работы оказывается принципиально ограниченной, а работоспособность зависит от наличия высокоскоростного доступа в интернет. Кроме того, передача изображений на сервер третьей стороны (не клиента и не банка, а компании, предоставляющей услуги по распознаванию), очевидно, не является безопасной схемой обработки данных. Чтобы избежать указанных недостатков, мы будем рассматривать задачу распознавания непосредственно на мобильном телефоне. Такие решения (например, программы card.io [3], InfoCard [4]) также представлены на рынке, однако их алгоритмы являются коммерческой тайной и не опубликованы.

Распознавание кредитной карты в полном объеме требует решения многих вопросов и использования алгоритмов из далеких друг от друга областей машинного зрения, решение этой проблемы вряд ли можно изложить подробно в пределах одной статьи. В данной работе мы ограничимся задачей поиска и распознавания даты окончания срока действия кредитной карты (далее просто даты). Подобные задачи (распознавание одного определенного информационного поля в документе) часто возникают при распознавании т.н. гибких форм [5], однако в нашем случае проблема усугубляется визуальными особенностями распознаваемого документа. Действительно, в общем случае кредитная карта является документом, обладающим формой с известным набором значащих элементов, причем на каждой отдельной карте некоторых из них может и не быть. При этом присутствующие элементы, их вид и положение на карте и друг относительно друга могут меняться. Таким образом, кредитная карта является типичным примером гибкой формы. В то время как распознавание документов с заранее известной и жестко фиксированной структурой достаточно хорошо изучено [6], существующие общие решения для гибких форм подразумевают распознавание всех возможных позиций и комбинаций символов и последующее составление осмысленного содержания исходя из полученного результата и предварительных знаний об объекте [7]. Данный подход неприменим в нашем случае, поскольку на кредитной карте допустимый фон настолько разнообразен, что не

удается построить достаточно надежный фильтр, выделяющий границы текстовых фрагментов а priori. Сказанное относится ко всем информационным полям карты, но именно у поля даты есть интересная особенность: поле «дата окончания срока действия» синтаксически неразличимо с часто присутствующим на карте полем «дата выдачи», что создает дополнительные трудности при распознавании и требует совместного анализа нескольких текстовых фрагментов.

Данная работа построена следующим образом. В главе 2 будет рассмотрена задача распознавания даты и ее место в более общей задаче распознавания кредитной карты в целом. В главе 3 будет рассказано об основных этапах распознавания информационных полей кредитной карты в рамках предлагаемого подхода. В главе 4 будут рассмотрены специфические проблемы распознавания поля даты и описаны способы их решения. В главе 5 даются и обсуждаются экспериментальные результаты применения предложенных алгоритмов.

2 Постановка задачи

В данной главе рассматривается постановка задачи определения местонахождения и распознавания даты окончания действия кредитных карт платежных систем «American Express», «VISA», «MasterCard», «DISCOVER», «Diners Club», «JCB» и «China UnionPay» на мобильных устройствах в режиме реального времени. Следует отметить, что в последнее время на рынок выходят новые национальные платежные системы (в частности, российская система ПРО100), но, по причине отсутствия незанятых ПИН-диапазонов, для международных платежей они представлены как одна из вышеперечисленных систем (т.н. кобрендинг).

Будем считать, что кредитная карта предъявляется фотокамере мобильного устройства под произвольным небольшим углом к плоскости изображения и полностью попадает в кадр хотя бы в некоторые моменты времени. Тогда четырехугольник наблюдаемых границ карты может быть найден по алгоритму, описанному в работе [8]. В реальном приложении пользователю во время съемки карты показывается найденная позиция четырехугольника, что позволяет ему интерактивно влиять на вероятность распознавания. Это, как и в случае аппаратов для считывания штрихкодов, позволяет улучшить эффективное качество распознавания в случае, если оператор (пользователь) в этом заинтересован.

При известном четырехугольнике границ нетрудно вычислить и применить к изображению проективное преобразование, приводящее изображение карты к ортогональному виду с фиксированным разрешением. Будем считать, что такое исправленное изображение и приходит на вход обсуждаемому алгоритму распознавания поля даты.

Результатом работы алгоритма должны быть 4 десятичные цифры: по две на месяц и год срока окончания действия. Считается, что алгоритм выдал правильный ответ, если полученные 4 цифры совпадают с теми, что изображены на

карте. Символ, разделяющий их, не учитывается и может быть любым. Отказ от распознавания трактуется как неверный ответ.

Кроме этого, скорость работы алгоритма должна быть достаточна для работы в реальном времени. Для оценки времени работы алгоритма будем использовать смартфон iPhone 4S, который по современным меркам является далеко не самым производительным устройством (для сравнения, более современный iPhone 6 быстрее в 7 раз согласно тесту Geekbench 3 Multi-Core Score). Вспомним, что излагаемый алгоритм является частью более сложной системы – приложения распознавания кредитной карты. Установим время полного распознавания в 1 секунду, поскольку сопоставимое время пользователь тратит на взаимное расположение документа и мобильного телефона, и такая задержка еще не будет вызывать раздражения. Для полного распознавания требуется провести поиск карты, распознавание номера, распознавание даты окончания действия и распознавание имени. Первые две задачи из перечисленных относительно просты, последняя – самая сложная. Считая нашу задачу средней по сложности, установим для распознавания даты лимит в 0.25 секунд на iPhone 4S.

3 Базовый алгоритм

Поиск поля даты «методом грубой силы» по всей площади карты – занятие малоперспективное, поскольку соответствующий текстовый фрагмент очень короткий (чаще всего 5 символов), синтаксическая избыточность невелика, и вероятность ложного обнаружения на произвольном фрагменте текста или даже пестром участке фона оказывается неприемлемо большой.

Воспользуемся следующим отсечением: будем искать дату только вблизи области возможного присутствия поля, определенной экспериментально на материале большого числа карт (более 10000 уникальных примеров). Это позволяет уменьшить площадь поиска даты примерно в 5 раз, тем самым уменьшив количество заведомо ложных срабатываний и увеличив скорость работы.

Опишем теперь последовательно шаги базового алгоритма.

Поиск информационных строк. Воспользуемся тем, что поле даты является частью информационной зоны, обладающей устойчивой геометрической структурой, причем расположение даты внутри этой зоны также хорошо определено. Рассматриваемая зона разбивается на три строки, так как большинство карт имеют именно три строки, одна из которых часто пустая. Эта парадоксальная фраза означает следующее: в случае, когда непустых строк в зоне две, их межстрочный интервал либо совпадает с межстрочным интервалом трехстрочных зон, либо примерно равен сумме удвоенного интервала и высоты строки (см. Рис. 1).

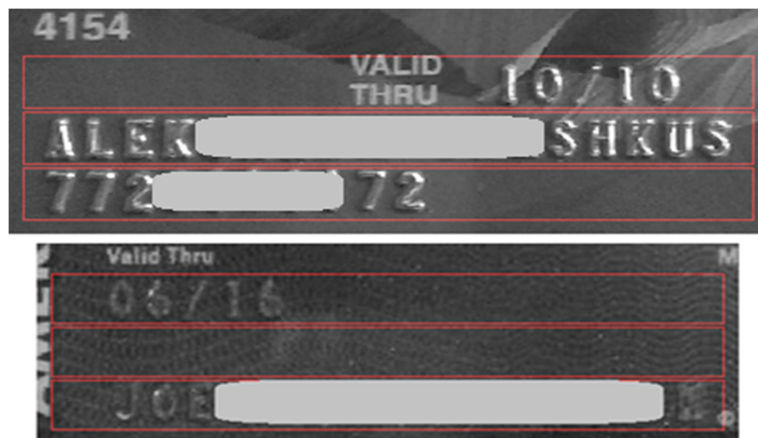


Рисунок 1. Трехстрочные зоны с полями под номером карточки: все три строки присутствуют (сверху), средняя строка пустая (внизу).

Поиск зоны и разбиение ее на 3 строки осложняется наличием на карте фона, который бывает удивительно разнообразным. Для решения этой задачи к области поиска применяется комбинация линейных и морфологических фильтров, целью чего является выделение вертикальных границ букв и гашение остальных деталей изображения. Для экономии времени морфологические операции реализованы с использованием алгоритма ван Херка [9].

После двумерной фильтрации интенсивности пикселей обработанного изображения проецируется на вертикальную ось.

$$p(y) = \sum_{x=0}^w \min(F(x, y), q_{\alpha}(F)) \quad (1)$$

Здесь F – результат фильтрации области поиска, w – ширина области, а $q_{\alpha}(F)$ – квантиль уровня α , близкого к 1, яркости F . Величина q используется для порогового отсечения с целью подавить влияние резких шумовых границ на исходном изображении. Как правило, это тоже текст, но не выдавленный, а нанесенный краской (на картах часто встречаются маленькие поясняющие надписи к полям, например “expires end” возле даты, см. Рис. 2), причем границы этого текста часто контрастнее, чем границы целевых символов.



Рисунок 2. Пример мелкого не выдавленного шрифта, сливающегося с символами даты.

По полученной проекции $p(y)$ теперь возможно найти наиболее вероятное положение строк, предполагая отсутствие горизонтальных границ в межстрочных интервалах. Для этого минимизируем сумму по проекции по всевозможным периодам h_l и начальным фазам y_0 из заранее заданного интервала.

$$\{H_l, Y_0\} = \operatorname{argmin}_{h_l, y_0} \sum_{i=0}^k p(h_l i + y_0) \quad (2)$$

Поскольку локальные минимумы p , как правило, достаточно выражены и на внешних границах текста, оптимальное значение k равняется четырём (подразумевая, что на карте 3 строки, а, следовательно, 4 локальных минимума). В результате мы найдем параметры $\{H_l, Y_0\}$, задающие центры межстрочных интервалов, а также внешние границы текста (см. Рис. 3).

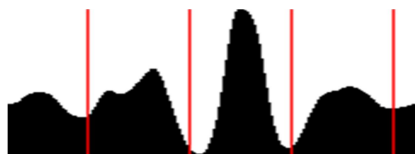


Рисунок 3. Вид проекции $p(y)$ и оптимальные разрезы, выделяющие на ней области трех строк.

Теперь область поиска даты можно существенно сократить, одновременно учитывая исходную форму области и найденное на данном изображении положение строк. Для такого пересечения генерируется множество возможных положений подстрок, с которым и будет вестись работа далее.

Распознавание подстрок. Каждая из подстрок-кандидатов сегментируется на символы, учитывая, что все символы на таких картах моноширинные. Это позволяет использовать алгоритм динамического программирования для поиска межсимвольных разрезов без распознавания символов [10]. После сегментации вырезанные изображения символов распознаются с помощью искусственной нейронной сети (ИНС) с алфавитом, содержащим цифры, знаки препинания, пробел и знак не-символа («мусор»). Таким образом, из множества возможных положений подстрок мы получаем множество строк, где каждый элемент строки – вектор псевдовероятностных оценок нахождения соответствующего символа алфавита в данной позиции строки.

Поиск даты. Для поиска наилучшего местоположения даты в такой строке применяется метод приложения шаблона. Наиболее часто встречающийся шаблон – «~ММ/YY~», где метасимвол «~» позволяет пробел или «мусор», а «М» и «Y» – любые цифры. Дублирующее обозначение позволяет указать, в каких знаках содержится искомая информация.

Для поиска оптимальной позиции максимизируется взвешенная сумма псевдовероятностей символов, подходящих под соответствующий метасимвол шаблона по всем возможным началам приложения шаблона d в строке.

$$d_m = \operatorname{argmax}_d \sum_{i=0}^n w_i \max_j s_{i+d,j} \delta(p_i, j) \quad (3)$$

Здесь w_i – вес позиции i в шаблоне, p_i – соответствующий метасимвол шаблона, $\delta(k, j)$ – функция проверки соответствия символа j метасимволу k , а $s_{i,j}$ – псевдовероятность нахождения символа j в позиции i .

Постобработка результата. После извлечения четырех искомым позиций из строки с наилучшим качеством приложения, к финальному результату применяется алгоритм контекстного уточнения результатов распознавания [11, 12], позволяющий исправить случайные ошибки ИНС исходя из ограничений на значения, ожидаемые в поле даты (например, не бывает 13-го месяца). Для этого используется т.н. алгоритм «фрулетки» [12], итерационно перечисляющий все возможные варианты прочтения подстроки в порядке убывания вероятности. Первый удовлетворяющий контекстным ограничениям вариант считается ответом.

Данная схема подходит для большого числа карт и уже в таком виде среднее качество распознавания даты составляет 81.83%.

Этот результат можно улучшить дополнительной обработкой. Считая позицию даты найденной в целом верно, можно достаточно надежно уточнить базовые линии символов даты, исключив влияние соседних областей. Уточненные позиции еще раз распознаются при помощи более сложной нейронной сети, имеющей лучшее посимвольное качество распознавания (использование ее на предыдущих этапах невозможно из-за ограничений по времени). После этого контекстная обработка повторяется, а достоверность ответов нейронной сети оценивается при помощи алгоритма «подсветки недостоверных символов» [13].

Если же мы желаем добиться чего-то существенно большего, нам необходимо рассмотреть специфику задачи детальнее.

4 Улучшения базового алгоритма

После реализации базовой версии алгоритма был проведен анализ ошибок на контрольной выборке данных объемом более 4000 изображений и составлен список основных проблем, которые необходимо решить для дальнейшего улучшения распознавания даты.

Оказалось, что сложности присутствуют на каждом этапе решения задачи. Рассмотрим наиболее значимые из них по отдельности.

Дата начала. Как показал анализ, на картах бывает указана не только дата окончания действия, но и дата выдачи. Базовый алгоритм не имеет инструмен-

тов для различения таких полей. В результате ответом считается то поле, которое с большей уверенностью распознано нейронной сетью. С точки зрения внешнего наблюдателя выглядит так, что из двух дат на карточке в качестве ответа выдается случайно выбранная.

Устойчивым свойством формы с двумя датами является расположение этих полей в одной строке, причем дата выдачи расположена левее (см. Рис. 4). Отсечение неверного варианта по абсолютным координатам не представляется возможным, поскольку в отсутствие даты выдачи дата окончания может располагаться в той же самой левой части информационной зоны.



Рисунок 4. Дата окончания срока действия находится левее даты выдачи на карте «American Express».

Казалось бы, при выборе можно опереться на безусловное упорядочение дат по времени. Оценим качество такого подхода. Система характеризуется вероятностью p ошибки распознавания поля. Будем считать p^2 пренебрежимо малым, а ошибки в четырех символах даты равновероятными. Тогда алгоритм выбора максимальной даты вернет заведомо ошибочный результат с вероятностью около $p/8$ в дополнение к вероятности ошибочного распознавания верного поля. Уточним, что установленного лимита времени хватает только на однократное распознавание поля даты «точной» ИНС ($p \approx 0.005$), поэтому сравнение результатов «грубого» распознавания ($p \approx 0.015$) приведет к окончательной оценке ухудшения финального качества в $11/8$ раз.

Поэтому мы ввели следующую проверку: если для лучшего приложения шаблона существует другое, такое, что они геометрически не пересекаются, относятся к одной строке, и лучший вариант – слева, то для альтернативного («правого») варианта качество приложения умножается на поправочный коэффициент, больший 1, что почти не дает дополнительных ошибок и не вызывает замедления системы.

Тип карты. Следующее улучшение связано с типом карты. Анализ более 10000 изображений различных карт показал, что возможные местоположения дат различны для карт разных платежных систем (на Рис. 5 изображена зона для карт «VISA»). Благодаря сокращению количества зон поиска даты можно добиться ускорения алгоритма и сокращения количества ошибок ложного обнаружения.



Рисунок 5. Зона поиска даты на примере карты VISA.

В общем случае распознавание даты само по себе не имеет смысла, так как для проведения транзакции необходим в первую очередь номер карты. Поэтому можно считать, что на момент начала распознавания даты в системе уже имеется распознанное поле номера, а по его значению можно однозначно установить платежную систему. Отметим, что качество распознавания номера превышает 99.65% (проверено на специальном выборке размером 6000 изображений), поэтому в дальнейшем мы будем доверять этому значению.

На данный момент исследованы варианты расположений полей на картах платежных систем «American Express», «VISA», «MasterCard», «DISCOVER», «Diners Club», «JCB» и «China UnionPay». В системе для каждого из типов карт рассматриваются области, описанные вокруг возможных положений даты.

Дополнительные паттерны. Следующий крупный кластер ошибок связан был с тем, что на картах периодически встречаются даты, выбивающиеся из формата ММ/YY, которые тоже нужно обрабатывать отдельным образом. К счастью, формула (3) легко обобщается на случай нескольких шаблонов. Максимум ищется не только по всевозможным приложениям шаблона, но и по всем шаблонам, а оценка нормируется на его длину.

В системе мы используем следующий список шаблонов даты: «~ММ+YY~», «~ММ/20YY~», «~###/##-ММ/YY~», «~ММ/##/YY~», «~ММ/~YY~». Здесь метасимвол «+» обозначает слеш или дефис, а «#» – любую цифру.

На Рис. 6 изображены фрагменты карт, иллюстрирующие разнообразие возможных внешних видов дат. Можно видеть, что и у даты начала, и у даты конца срока действия встречаются разные шаблоны.



Рисунок 6. Примеры дат с различными шаблонами.

Пропуск строки. Отдельно стоит отметить ряд карт, которые имеют две строки с нестандартным межстрочным интервалом. На Рис. 7 приведен пример зоны карты под ее номером, на которой видно, что расстояние между строкой с цифрами и именем больше обычного (показанного на Рис. 1).

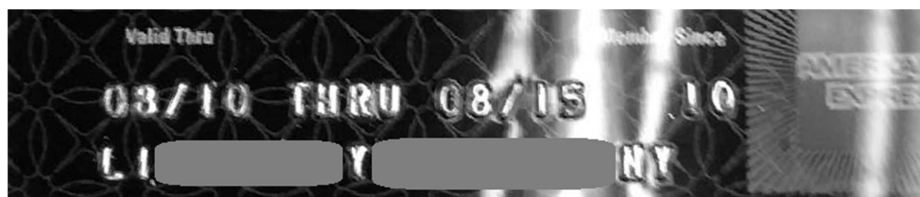


Рисунок 7. Нестандартное расстояние между строками.

При разбиении на строки стандартным алгоритмом одна из двух строк будет исключена из дальнейшего анализа, что сильно повышает вероятность ошибки. Для корректной обработки этого типа карт была использована следующая модификация алгоритма поиска строк. В случае, если на проекции p изображения зоны даты присутствуют ровно два сильно выраженных минимума, то именно они считаются центрами строк.

5 Результаты

В данном разделе представлены результаты применения алгоритма и различных его модификаций. Полный алгоритм решения задачи распознавания даты реализован в виде приложения. Время работы программы на устройстве iPhone 4S занимает 0.2 секунды, что позволяет применять его на мобильном устройстве, а не передавать изображение для распознавания на сервер. Сравнительное качество системы с различными модификациями предоставлено в таблице 1.

Таблица 1. Точность итогового распознавания.

Версия	Точность с одной ИНС	Точность с комбинацией ИНС
Базовая версия	81.83	82.75
Базовая + пропуск строки	82.20	83.17
Базовая + дополнительные паттерны	87.01	87.99
Базовая + тип карты	89.47	90.28
Базовая + дата начала	94.35	95.44
Полная версия	98.56	99.51

Проверка качества алгоритма производилась на специальном предварительно размеченном наборе данных. Набор состоит из 4320 изображений уникальных карт. При отборе картинок критерием того, что система должна распознавать картинку был факт читаемости даты человеком: если человек способен прочесть (не разобрать, используя приближения и различные программные фильтры) дату на изображении – значит, это изображение должно распознаваться.

В Таблице 1 показан эффект воздействия каждой модификации на качество распознавания по отдельности, чтобы вклад каждой из компонент можно было оценить по степени значимости.

6 Заключение

В статье была рассмотрена задача распознавания гибкой формы на примере распознавания даты окончания срока действия кредитных карт платежных систем «American Express», «VISA», «MasterCard», «DISCOVER», «Diners Club», «JCB» и «China UnionPay». Рассмотрены основные проблемы, возникающие при решении поставленной задачи. Предложен и программно реализован подход к решению. Даны результаты применения данного подхода и представлены численные результаты сравнения с и без применения улучшений базовой версии алгоритма.

Предложенное решение соответствует высоким требованиям по качеству распознавания и скорости работы, что делает возможным его применение в действующей системе распознавания кредитных карт. В дальнейшем планируется ускорение и улучшение алгоритма путем использования в нем меньшего числа более универсальных подсистем. Кроме этого, кажется перспективной задача ускорения работы нейронных сетей, что позволило бы применить финальное распознавание к большему числу вариантов и улучшить качество, не потеряв в скорости.

Работа выполнена при частичной финансовой поддержке РФФИ (проекты 13-07-12172-офи_м, 13-07-12173-офи_м).

Список литературы

1. Jean Lassignardie, Kevin Brown. World Payments Report 2014 // Capgemini and RBS, September 16, 2014, pp. 1 - 60.
2. Wang X., Bissacco A., Berntson G., Nazif M., Scheiner J., Shih S., Snyder M. L., Talavera D. Client side filtering of card OCR images // Патент США US8903136 B1, 02.12.2014.
3. URL: <https://www.card.io/> (дата обращения: 04.08.2015).
4. URL: <http://www.infodif.com/infocard/> (дата обращения: 04.08.2015).
5. Постников В.В. Автоматическая идентификация и распознавание структурированных документов : автореферат дис. кандидата технических наук : 05.13.01 / Ин-т систем. анализа РАН. - Москва, 2001. - 16 с.
6. Усилин С.А., Николаев Д.П., Постников В.В. Локализация, ориентация и идентификация документов с фиксированной геометрией на изображении // Труды Института системного анализа РАН. Обработка информационных и графических ресурсов / под ред. В.Л. Арлазаров. М.: КРАСАНД, 2010. С. 248–261.
7. Постников В.В., Марченко А.Е., Шоломов Д.Л. Разбор структурированного документа в модели с нечеткой логикой // Труды Института системного анализа РАН. Документооборот. Концепции и инструментарий. М.: URSS, 2004, стр. 71-82.
8. Skoryukina N., Nikolaev D., Sheshkus A., Polevoy D. Real time rectangular document detection on mobile devices // Proc. SPIE 9445, Seventh International Conference on Machine Vision (ICMV 2014), 2015, 94452A, pp. 1-5.
9. van Herk M. A fast algorithm for local minimum and maximum filters on rectangular and octogonal kernels // Pattern Recognition Letters, 1992, V. 13, № 7, pp. 517–521.
10. Куроптев А.В., Николаев Д.П., Постников В.В. Точная локализация опорных решеток полей заполнения в анкетах методами динамического программирования и морфологической фильтрации // Труды Института системного анализа РАН, 2013, Т. 63, № 3, стр. 111-116.
11. Sholomov D.L. Syntactical Approach to Post-Processing of Fuzzy recognized Text // Proc. of The International Conference on Machine Learning, Technologies and Applications. CSREA Press, June 2003, USA. P. 115–121.
12. Арлазаров В.В., Малых В.А., Шоломов Д.Л. Распознавание изображений документов с использованием алгоритма «рулетки» // Труды Института системного анализа РАН, 2013, Т. 63, № 4, стр. 35-38.
13. Арлазаров В.В., Булатов К.Б., Карпенко С.М. Метод определения надежности распознавания тисненых символов // Труды Института системного анализа РАН, 2013, Т. 63, № 3, стр. 117-122.