# Формирование множества графических образов символов в задачах обучения классификатора символов*

В. В. Арлазаров, Н. В. Решетняк, О. А. Славин

**Аннотация.** В работе описываются задачи создания обучающего множества образов символов в условиях ограничения времени работы операторов, проводящих проверку и разметку образов символов. Предложен метод формирования обучающего множества образов символов (как бинарные образы, так и образы серых и цветных), основанный на разделении процесса проверки результатов распознавания между операторами, контролирующими распознавание текстовых строк, и операторами, выполняющими окончательную разметку.

**Ключевые слова:** *изображение, распознавание символов, образ символа, обучающее множество, обучающий пример.*

## Введение

В современном мире широко представлены системы перевода документов в электронный вариант. Примерами могут служить распознавание структурированных документов (анкет, паспортов, водительских удостоверений), распознавание текстовых документов с сохранением форматирования (журналы, газеты) или оцифровка книг. В основе всех алгоритмов распознавания текстовых документов лежат *классификаторы*, которые предлагают набор альтернатив принадлежности образа символа нескольким классам с оценками надежности. Результатом классификации может быть отнесение образа символа к классу несимволов $\Delta$.

Качество распознавания документов напрямую зависит от качества распознавания символов, следовательно, особое место занимает задача обучения классификатора. Начальным этапом построения классификатора служит формирование обучающего множества, состоящего из образов символов различного вида (черно-белых (бинарных), полутоновых, цветных) и соответствующих символам атрибутов (код символа в некотором алфавите, признаки шрифта (жирность, курсивность, гарнитура)). Совокупность атрибутов определяет *алфавиты* обучения и классификации. Результативность обучения классификатора, т. е. достижения высокой точности распознавания и монотонность оценок надежности распознавания, сильно зависит от объема обучающего множества и от точности соответствия установленных атрибутов символов.

Вообще говоря, все написанное относится и к тестовому множеству, необходимому для проверки качества обучения классификатора, далее мы будем рассматривать только обучающие множества.

Процесс обучения рассматривается как итерационный, т. е. для первичного обучения и последующих сеансов дообучения используются различные обучающие множества.

При построении обучающего множества нужно особое внимание обратить на репрезентативность данных. В работе [1] уделено внимание основным принципам формирования обучающего множества:

- *достаточность* — число обучающих примеров должно быть достаточным для надежного обучения. Разумеется, число примеров может быть различным для разных моделей обучения. Например, для нейронной сети необходимо, чтобы число обучающих примеров было в несколько раз больше, чем число весов межнейронных связей, в противном случае модель может не приобрести способности к обобщению [1]. В реальности достаточность оценивается с помощью характеристик обученного метода, например, исследованием зависимости точности распознавания от числа обучающих примеров в предположении, что график этой зависимости монотонен.
- *разнообразие* — большое число разнообразных возможных комбинаций признаков в обучающих

* Работа выполнена при поддержке РФФИ (проект № 13-07-12170).

примерах. Этот принцип тесно связан с предыдущим, он усиливает требования к числу обучающих примеров, в которых явно оцениваются используемые в классификаторе признаки и их комбинации. Оценка разнообразия может быть проведена с помощью кластеризации, использующей представления образа в виде набора признаков, разнообразие оценивается, например, зависимостью числа получившихся кластеров от числа обучающих примеров.

– *распределение частот классов* — примеры различных классов должны быть представлены в обучающей выборке примерно в пропорциях, соответствующих пропорциям классов в тестовой выборке. Преобладающие классы будут определены как более вероятные для новых наблюдений. При создании классификатора стандартных текстов определенного языка разумно ориентироваться на частотность распределения встречаемости отдельных символов [8].

Набор образов символов для обучения классификатора может быть сформирован различными способами: сгенерирован искусственно, извлечен из изображений (тех, которые будут распознаваться, или из похожих изображений). Далее необходима *разметка* образов символов, состоящая в приписывании каждому образу его атрибутов, как минимум кода символа. Разметка может проводиться как автоматически, так и вручную, при создании классификаторов для распознавания документов, ручная разметка обязательна, если требуется обучение на примерах, извлекаемых из последовательности случайных образов документов. Ручная разметка может быть основана как на предъявлении *оператору* (лицу, осуществляющему разметку) отдельного символа, так и на предъявлении части образа документа с контекстным окружением этого символа. Последний способ разметки является более точным, нежели первый, но он требует больше времени оператора на анализ каждого символа.

Задача формирования обучающего множества образов символов состоит в получении как можно больше надежно размеченных разнообразных образов. Другими словами, задача сводится к следующим:

– оценка объема множества и, возможно, оценка количества комбинаций признаков;
– минимизация ошибочно размеченных образов, а также оценка доли ошибочно размеченных образов;
– оценка соответствия распределения частот классов заранее заданному распределению.

Необходимо учесть, что автоматическая разметка не позволяет создать обучающие множества, репрезентативные для большого набора произвольных изображений, а ручная разметка требует ресурсов операторов.

В случае применения ручной разметки возникает другая задача: как при имеющихся ресурсах операторов за ограниченное время создать обучающее множество наибольшего объема?

Исследованию этих задач посвящена данная статья.

## 1. Обзор существующих методов построения обучающего множества

Задача построения множества образов символов может быть довольно трудоемкой и затратной [2, 3]. В процессе формирования реальных данных приходиться решать подзадачи предварительной обработки, такие как поиск символов, удаление шумов и посторонних объектов. Сложность построения множества образов отдельных символов отмечена в работе [4], в которой отмечается, что качество обучающего множества напрямую зависит от точности алгоритма поиска границ символа: каждый символ должен быть строго центрирован, одни и те же символы должны иметь одинаковые размеры.

В работах [2, 3] указано на создание специализированных форм, при заполнении которых необходимо придерживаться определенных правил. Указанный подход к формированию обучающего множества требует больших человеческих затрат. На этапе заполнения формы, как было отмечено ранее, важно получить как можно более широкий диапазон вариантов написания каждого класса образов. Эта особенность требует большего числа респондентов, что делает этот подход к созданию обучающего множества дорогим и неэффективным.

В работе [5] рассмотрены различные способы получения базы графических символов. Отмечено явное преимущество сохранения образов символов непосредственно из программы распознавания документов. Например, при такой схеме создания множества образов достаточно точно определяются границы каждого символа. Естественно, возникает вопрос о надежности распознавания отдельного символа.

## 2. Модель процесса формирования обучающего множества

Рассмотрим модель процесса формирования обучающего множества большого объема.

Пусть существует источник образов, из которого поступают как образы символов, так и образы несимволов.

Рассмотрим два механизма разметки образцов символов, поступающих из некоторого источника:

автоматический классификатор образов (OCR) и разметка операторами образов, которые могут быть как предварительно классифицированными, так и не классифицированными.

Автоматическая разметка проводится быстро, но сопряжена с ошибками. Ручная разметка является более точной, но ограничена скоростью работы операторов.

Рассмотрим подробнее процедуру разметки набора образов символов. Разметка включает в себя следующие операции:

- изменение образа символа;
- изменение границ символа;
- проверку границ символа с возможной *отбраковкой* (удалением образа из обучающего множества);
- ввод кода символа;
- проверку кода символа с возможной отбраковкой.

Приведенные операции упорядочены по убыванию затраченного на операцию времени. Время на выполнение операции проверки кода символа с возможной отбраковкой может быть уменьшено, если оператору подаются на разметку ранее классифицированные символы с одинаковым кодом. Необходимо отметить, что функция отбраковки уменьшает время работы оператора, но в то же самое время уменьшает разнообразие признаков в образах обучающего множества.

Как уже отмечалось выше, оператору может предъявляться как отдельный образ, так образ в контексте соседних символов, в последнем случае *точность* разметки, определяемая как отношение количества ошибочно размеченных образов к общему количеству образов, повышается за счет увеличения расхода времени оператора на анализ группы символов.

Время выполнения операций варьируется от 0,3–0,5 секунды для операции проверки кода символа с возможной отбраковкой (в случае, когда предъявляются однородные образы, которые заранее отсортированы по коду символа и иным признакам) до 30 и более секунд для операции изменения образа символа.

Рассмотрим разбиение множества $M$, извлеченного из некоторого источника образов, на 3 подмножества $M_a \cup M_v \cup M_e$, где

- $M_a$ — множество уверенно классифицированных образов символа — эти образы не подлежат ручной проверке.
- $M_v$ — множество образов, требующий ручной проверки, во-первых, факта принадлежности к символам, и, во-вторых, правильности классификации.
- $M_e$ — множество образов, которые не могут быть классифицированы автоматически и которые оператор при ручной проверке должен классифицировать заново.

Зададимся оценками времен $t_v$ и $t_e$ обработки одного образа оператором из множеств $M_v$ и $M_e$, соответственно.

Тогда общее время обработки оператором множества $M$ определится как

$$t = |M_v| \cdot t_v + |M_e| \cdot t_e.$$

Способ разбиения множества $M$ задаст время обработки $t$. Для больших объемов множеств время обработки почти всегда ограничено. Например, для $|M| = 1\,000\,000$ образов, при $t_e = 0{,}5$ сек, общее время обработки каждого символа составит примерно 18 дней. Отсюда следует, что для описанной работы не удастся ограничиться одним оператором, и что необходимы средства автоматизации процесса формирования обучающего множества.

Нередко возникают затруднения при классификации похожих символов, например, необходимость различать буквы «О» и цифры «0». Для решения этой проблемы необходимо обратиться к образу текстового поля, из которого был получен символ. Оператору должна быть доступна исходная текстовая строка с указанием текущего символа в текстовом поле. Таким образом, контекст поля существенно повышает качество обучающего множества.

Из вышесказанного следует, что способ разбиения множества $M$ на подмножества $M_a$, $M_v$, $M_e$ и способ представления элементов этих множеств позволяют минимизировать время, затраченное на обработку оператором подмножеств $M_v$, $M_e$, и минимизировать количество ошибок классификации образов множества $M$. Отметим, что способ разбиения множества $M$ на подмножества также может включать ручные операции, которые необходимо учесть при оценке общих затрат времени.

## 3. Способ формирования множества образов символов в процессе эксплуатации OCR-системы

В задаче распознавания документов, например, при сохранении в архиве потока образов документов, ошибочно распознанные образы должны быть исправлены или, как минимум помечены как ненадежно распознанные. Этапы верификации и редактирования результатов распознавания обусловлены бизнес-логикой системы распознавания документов [10]. Эти этапы проводятся силами операторов из организации, эксплуатирующей OCR-систему.

Достаточно часто результаты распознавания документов, то есть документы в цифровом виде,

не могут быть переданы разработчикам OCR-системы из организации, эксплуатирующей OCR-систему, прежде всего, по требованиям информационной безопасности. Однако результаты распознавания, состоящие из множества $M_a \cup M_v \cup M_e$, не позволяют восстановить исходные документы, и могут быть переданы разработчикам OCR-системы для повторного обучения классификаторов.

То есть процесс разбиения на $M_a \cup M_v \cup M_e$ осуществляется на технических средствах организации, эксплуатирующей OCR-систему. Несмотря на использование для разбиения результатов верификации и редактирования, операторам не приходится делать никаких новых специальных действий.

Предлагаемый нами способ формирования множества образов символов основан на использовании результатов распознавания текстовых полей и тестовых строк, подтвержденных оператором.

## 4. Алгоритмы извлечения множества образов символов из распознанных строк при редактировании и верификации

Рассмотрим задачу посимвольного сопоставления результата распознавания строки (набор альтернатив с весами для каждого образа) и соответствующей последовательностью символов, подтвержденной оператором на этапах редактирования и верификации. То есть каждому символу текстовой строки нужно соотнести образ символа распознаваемой строки.

Решение задачи, то есть сопоставление результата распознавания с набором символов, будем производить методом динамического программирования с метрикой Левенштейна [6]. Возьмем за основу базовые принципы алгоритма MCHSR [7]. MCHSR является одним из методов контекстной обработки результатов распознавания. Этот алгоритм был разработан для поиска вхождения фрагмента текста в строке результатов распознавания. Мы же рассматриваем задачу полного наилучшего сопоставления подтвержденной строки с результатом распознавания.

На первом шаге алгоритма построим таблицу, в ячейках которой будет указано качество классификатора символа, если символ совпадает с одной из альтернатив для текущего знакоместа. Если текущий символ отсутствует в списке альтернатив, то клетку таблицы оставим пустой.

На втором шаге алгоритма найдем наилучший путь (путь наибольшего веса) из левой нижней точки таблицы (синяя точка) в правую верхнюю точку (зеленая точка). Разрешены следующие переходы:

– вверх по ребру ячейки таблицы — случай, когда среди результатов распознавания отсутствует альтернатива, соответствующая введенному оператором символу (образ символа был не распознан). Для простоты изложения будем считать стоимость перехода равную 0.
– вправо по ребру ячейки таблицы — случай, когда результату распознавания не соответствует ни один из символов строки (фрагмент мусора распознан как символ). Аналогично, будем считать стоимость перехода равную 0.
– переход по диагонали ячейки — сопоставление символа с одной из альтернатив знакоместа. Стоимость перехода положим равной значению ячейки.

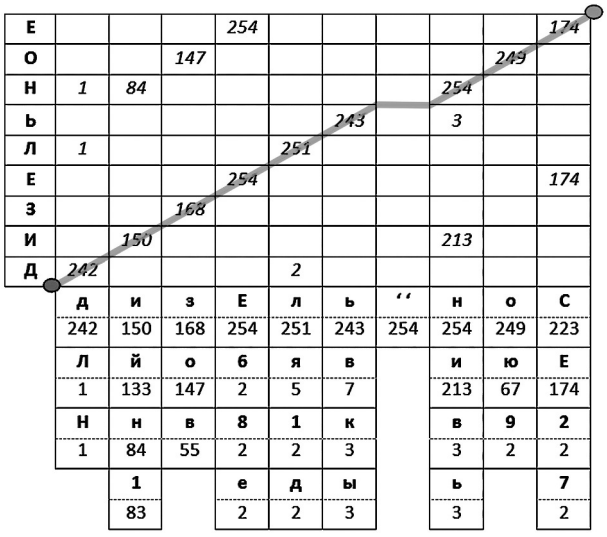


**Рис. 1.** Модель сопоставления результата распознавания и текстовой строки, введенной оператором: построение таблицы и поиск лучшего соответствия растровых образов и символов (указано жирной серой линией)

В результате вышеописанного алгоритма будет сформирован набор соответствий: символ тестовой строки — образ символа. Возможны случаи, когда для символа не найден растровый образ и, наоборот, для растрового образа не найден символ.

После сопоставления каждый растровый образ можно отнести к одному из трех видов:

– уверенно распознанный образ символа — для данного образа наилучшая альтернатива символа (альтернатива с наибольшим весом) соответствует символу из строки и сама альтернатива имеет высокое качество распознавания;
– образ, требующий подтверждения — для данного образ символа одна из второстепенных альтернатив (не наилучшая альтернатива) соответствует символу из строки;
– «неправильно» распознанный образ символа — образ символа, для которого не найден соответствующий символ из строки.

Таким образом, мы получили три множества образов символов $M_a$, $M_v$, $M_e$.

Сохраняемые символы могут быть представлены как бинарными, так и полутоновыми и цветными образами. В последних случаях для обучения могут понадобиться не только образы как таковые, но и параметры отделения полутоновых и цветных образов букв от фона. В простейшем случае порог отделения фона может быть взят из результатов бинаризации группы символов, составляющих строку, и уточнен адаптивными алгоритмами расчета порога бинаризации, например, метод Ниблэка [9].

## 5. Экспериментальное исследование разбиения обучающего множества на части для оценки времени разметки

В данном разделе описан эксперимент получения и разделения образов символов на три группы. Было получено более 2 000 000 символов, среди которых более 82 % образов были отнесены к множеству $M_a$. Доля символов, требующая дополнительную проверку оператором, составила менее 10 %, что позволяет существенно ускорить процесс создания обучающего множества. Множество символов $M_e$ составило около 8 %.

На этапе отнесения образа символа к одной из трех групп предложено считать символ надежно распознанным, если первая альтернатива имеет высокое качество. Мы исследовали размер множества образов $M_v$ от зафиксированного качества символа первой альтернативы $Q_0$. Иными словами, символ не нужно дополнительно подтверждать, если качество первой альтернативы $q > Q_0$, иначе символ попадает в множество сомнительно распознанных образов. На рис. 3 представлены результаты эксперимента для документов «паспорта РФ».

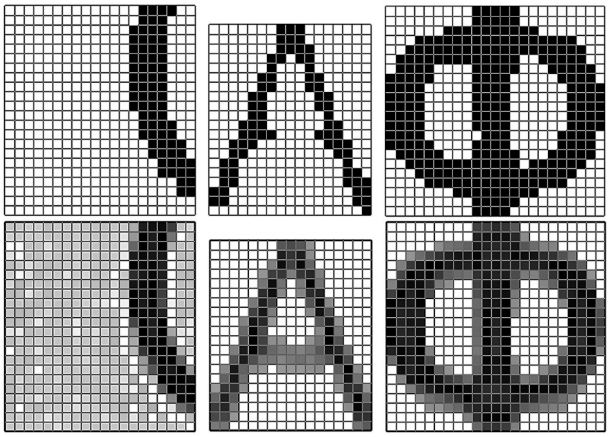

**Рис. 2.** Примеры символов из разных множеств. **Слева:** пример образа, не являющегося символом; справа: символ из множества уверенно распознанных образов; в центре: образ символа, требующий дополнительную проверку оператором

Центральное место в задаче формирования обучающего множества занимает надежность классификации образов. Нами была изучена зависимость количества ошибок множества $M_a$ от качества символа первой альтернативы. Мы подсчитали количество ошибок для каждого промежутка значений качества альтернативы на стенде «Паспорта РФ». На рис. 4 видно, что процент ошибок уменьшается с ростом значения альтернативы.

Возникает вопрос, можно ли создать множество $M_a$, где доля ошибочно классифицированных образов не более заранее заданного числа $p$?

Исследование возникшей проблемы показало (рис. 5), что с ростом первой альтернативы $Q_0$ процент ошибочно классифицированных образов в мно-


**Рис. 3.** Зависимость размера (в % от общего числа образов $M$) базы $M_v$ от минимально допустимого значения качества образа из множества $M_a$ для паспортов РФ

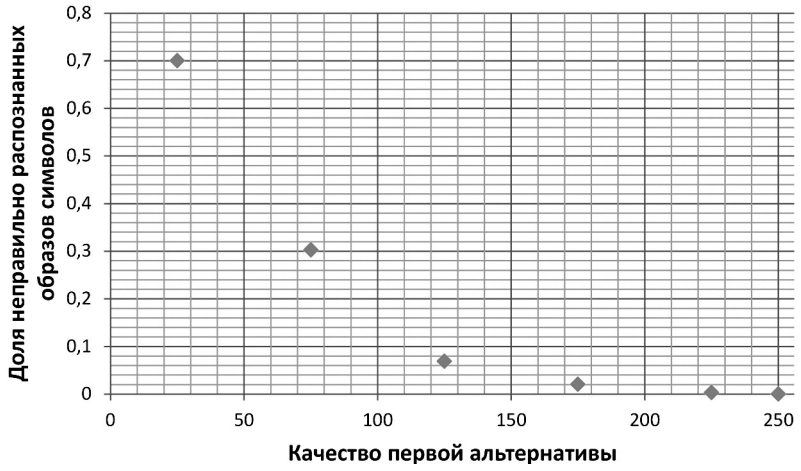


**Рис. 4.** Зависимость количества неверно распознанных символов от качества первой альтернативы

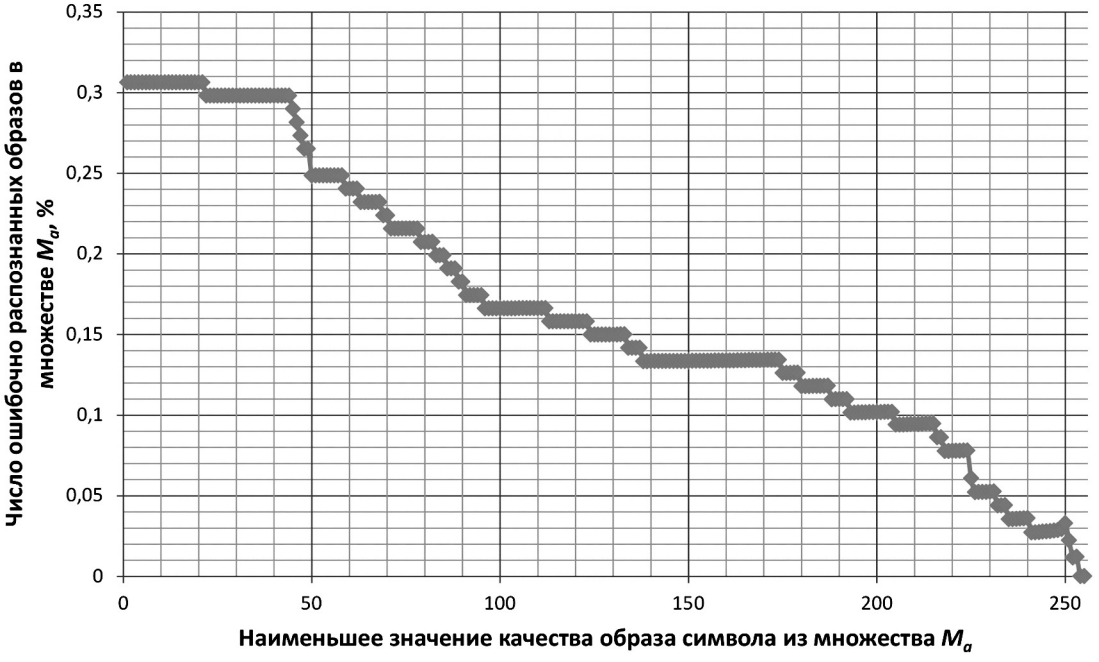


**Рис. 5.** Зависимость процента ошибочно распознанных образов в множестве $M_a$ от качества символов, составляющих это множество

жестве $M_a$ монотонно убывает. Таким образом, для заданного числа ошибочно классифицированных образов $p$ найдется значение $Q_0$, при котором множество $M_a$ содержит менее $p$ ошибочно классифицированных образов.

Посчитаем требуемое время на создание множества графических образов предложенным способом, и сравним его с временем классификации каждого символа множества $M$. Будем производить расчет для образов символов, полученных на стенде «Паспорта РФ». Предположим, необходимо разметить 1 000 000 образов символов, с долей ошибочно классифицированных символов не более 0,1 %. По изложенным выше расчетам, разметка всех символов составит 18 дней.

Множество $M_a$ будет содержать менее 0,1 % ошибочно классифицированных образов при $Q_0 = 200$ (рис. 5). Для полученного значения $Q_0$, размер множества $|M_v|$ составит 100 000 образов (рис. 4). Следовательно, время разметки 1 000 000 образов символов составит около $T = |\mathrm{M}_v| \cdot t_v \approx 2$ дня.

Не будем забывать, что реальный оператор не может работать 8 часов в день с одинаковой производительностью, что приведет к пропорционально-

му увеличению затрат времени на разметку обоими способами.

Расчет показал, что предложенный метод позволяет ускорить процедуру формирования обучающего множества более чем в 9 раз.

## Вывод

В работе предложен метод формирования обучающего множества образов символов (как бинарные образы, так и образы серых и цветных), основанный на разделении процесса проверки результатов распознавания между операторами, контролирующими распознавание текстовых строк, и операторами, выполняющими окончательную разметку.

Были проведены эксперименты с использованием программы распознавания документов Cognitive Forms, в которых было исследовано число ошибок в обучающем множестве и размер множества, требующего дополнительной проверки оператором $M_v$, от качества образов символов, формирующих набор уверенно распознанных символов $M_a$.

На примере паспортов РФ было показано и экспериментально проверено преимущество изложенного метода. В результате было сформировано обучающее множество, состоящее более чем из 2 000 000 образов символов. Время на классификацию образов было затрачено в 9 раз меньше, чем при непосредственной разметки каждого символа множества. Предложенный способ позволяет уменьшить число проверяемых образов без существенного ущерба качеству формируемого множества, тем самым уменьшить время создания размеченного множества реальных образов символов.

Также отметим, что предложенный способ позволяет совершенствовать классификатор системы OCR, используемой в некоторой организации, в основном, за счет разметки, предусмотренной регламентом работы операторов в процессе этой системы.

## Литература


1. *Галушка В. В., Фатхи В. А.* Формирование обучающей выборки при использовании искусственных нейронных сетей в задачах поиска ошибок баз данных // [Электронный ресурс], Инженерный вестник Дона. 2013. № 2. http://www.ivdon.ru/magazine/archive/ n2y2013/ 1597 (дата обращения: 06.08.2013).
2. *Huda Alamri, Javad Sadri, Ching Y. Suen, Nicola Nobile.* A Novel Comprehensive Database for Arabic Off-Line Handwriting Recognition // Proc. of the 11th Int. Conference on Frontiers in Handwriting Recognition (ICFHR' 2008). Montreal, Canada. 2008. P. 664–669.
3. *Tonghua Su, Tianwen Zhang, Dejun Guan.* HIT-MW dataset for offline Chinese handwritten text recognition. In 10th IWFHR, La Baule, 2006.
4. *Савчинский Б. Д., Олефиренко С. А.* Поиск размеров эталонов при распознавании текстовых изображений // Сборник трудов Международного научно-обучающего центра информационных технологий и систем НАН и МОН Украины «Перспективные технологии обучения и учебных центров». К.: МННЦИТИС, 2009. Вып. 2. С. 24–45.
5. *Славин О. А.* Средства управления базами графических образов символов и их место в системах распознавания // Сборник трудов ИСА РАН «Развитие безбумажных технологий в организациях». М.: URSS, 1999. С. 277–289.
6. *Левенштейн В. И.* Двоичные коды с исправлением выпадений, вставок и замещений символов // Доклады Академий Наук СССР. 1965. Т. 163. №. 4. С. 845–848.
7. *Постников В. В.* Автоматическая идентификация и распознавание структурированных документов // Дисс. на соискание уч. ст. канд. техн. наук. 2001.
8. *Яглом А. М., Яглом И. М.* Вероятность и информация. М.: КомКнига/URSS, 2007.
9. *Wayne Niblack.* An Introduction to Digital Image Processing. Englewood Cliffs, Prentice Hall, N. J., 1986. P. 115–116.
10. *Арлазаров В. В., Постников В. В., Шоломов Д. Л.* Cognitive Forms — система массового ввода структурированных документов // Сборник трудов Института системного анализа РАН «Управление информационными потоками». М.: URSS, 2002. С. 35–46.



**Арлазаров Владимир Викторович.** Зав. лабораторией ИСА РАН. К. т. н. Окончил в 1999 г. МИСиС. Количество печатных работ: 11. Область научных интересов: распознавание образов, обработка изображений, системы массового обслуживания. E-mail: vva777@gmail.com

**Решетняк Никита Валерьевич.** Студент магистратуры МФТИ. Окончил в 2014 г. бакалавриат МФТИ. Область научных интересов: распознавания образов. E-mail: nikitaresh@cs.isa.ru

**Славин Олег Анатольевич.** Зав. лабораторией ИСА РАН. Д. т. н. Окончил в 1988 г. Московский институт радиотехники, электроники и автоматики. Количество печатных работ: 69 (в т. ч. 1 монография). Область научных интересов: распознавание образов, информационные системы. E-mail: oslavin@cs.isa.ru