ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования

Национальный исследовательский университет
«Высшая школа экономики»

Факультет гуманитарных наук
Образовательная программа
«Фундаментальная и компьютерная лингвистика»

Мустакимова Эльмира Гаязовна

РАСПОЗНАВАНИЕ РУССКОГО РУКОПИСНОГО ТЕКСТА

Выпускная квалификационная работа
студента 4 курса бакалавриата группы БКЛ121

Академический руководитель
образовательной программы
канд. филологических наук, доц.
Ю. А. Ландер
_____
«___» _____ 2016 г.

Научный руководитель
канд. филологических наук, доц.
О. Н. Ляшевская
_____
«___» _____ 2016 г.

Москва 2016

Оглавление

Распознавание русского рукописного текста

Эльмира Мустакимова

Аннотация

Современные исследования в распознавании русского рукописного текста фокусируются на печатном или рукопечатном шрифте. Лишь небольшое число распознающих систем могут хорошо анализировать курсивный русский текст. Однако все такие системы разрабатываются в коммерческих целях и закрыты для научного сообщества. В этой работе мы ставим целью разработать открытую систему OCR (Optical Character Recognition) для распознавания русского курсивного письма. В разработке этой системы используются нейронные сети и метод опорных векторов. Также в рамках этой работы мы создали базу данных с рукописными символами и целыми предложениями на русском языке. Эта база данных позволит сравнивать между собой результаты разных систем распознавания русского рукописного текста.

1. Введение

Люди, в отличие от компьютеров, могут легко читать рукописный текст: мы можем различать буквы знакомого алфавита, узнавать целые слова и понимать предложения вне зависимости от конкретного почерка. Современные компьютеры могут только достаточно точно распознавать большинство шрифтов печатного текста. Некоторые системы OCR (Optical Character Recognition) показывают хорошие результаты и на рукопечатных текстах (т.е. текстах, написанных от руки печатными буквами), но совсем небольшое число систем могут верно распознавать со сканированного изображения курсивный рукописный текст. Таким образом, оффлайн-распознавание рукописного текста все еще является сложной задачей для компьютеров. Под оффлайн-распознаванием мы понимаем выделение текста из отсканированной рукописи, например, со сканированной копии письма, вручную заполненной анкетной формы или письменного экзамена. Оффлайн-распознавание противопоставляется онлайн-распознаванию, где текст извлекается непосредственно в процессе письма на сенсорном устройстве. До сих пор не существует свободно распространяемых систем для оффлайн-распознавания русского рукописного текста.

Создание подобной системы OCR требует большого количества примеров рукописного письма и разных почерков, на которых можно было бы натренировать систему. Подобные примеры были собраны в обширные базы данных для многих языков и письменностей. Такие базы данных могут содержать аннотированные изображения рукописных букв или цифр, слов или предложений и иногда могут предоставлять информацию об авторах, например, пол писавшего, возраст, город или же просто индивидуальный номер для того, чтобы отличать разных авторов друг от друга. Для русского языка существует подобная открытая база данных, содержащая примеры рукописных букв русского алфавита и цифр (Истратов & Федин

2013). Однако, до сих пор нет открытой базы рукописных слов и предложений, на которой можно было бы обучать системы распознавания текстов, а не отдельных символов.

Таким образом, в этой работе мы ставим перед собой две большие задачи. Во-первых, мы создаем базу рукописных слов и предложений на русском языке, и во-вторых, мы разрабатываем свободно распространяемую систему OCR для извлечения текста из изображения русского курсивного письма.

Эта система могла бы использоваться для распознавания текста в рукописных архивах писателей (старые дневники, переписка, черновики), для автоматической обработки заполненных форм, для распознавания адресов на почтовых конвертах или же для выделения текста из рукописных конспектов.

В нашей системе OCR используются нейронные сети и метод опорных векторов для распознавания отдельных букв, а также словарь – для формирования слов и предложений на основе предсказания нейронной сети. Созданная нами база данных доступна онлайн (`http://elmiram.github.io/ruhwr/`) и может использоваться для разработки и тестирования систем OCR, подобных нашей, а также для сравнения разных систем на одном и том же наборе данных. Наша база данных и разработанная система являются вкладом в исследования в области компьютерного зрения.

В следующем разделе мы опишем результаты предшествующих работ, а также существующие базы данных для распознавания рукописного текста. Раздел 3. описывает стоящую перед нами задачу. В разделе 4. мы описываем формирование и структуру нашей базы данных рукописных предложений. В разделе 5. описаны эксперименты по распознаванию символов, слов и предложений, результаты этих экспериментов и сравнение с другими работами. Наконец, в разделе 6. мы подводим итоги выполненной работы и формулируем дальнейшие задачи.

2. Предшествующие работы

2.1. Стандартные базы данных

Сообщество исследователей в области распознавания рукописного текста разработали большое количество баз данных для оценки OCR-систем. Базы данных для оценки оффлайн-распознавания рукописного текста доступны для многих языков и систем письменностей. Например, английский рукописный текст представлен в базе данных IAM (Marti & Bunke 2002) и CEDAR (Hull 1994). Французский представлен в базе RIMES (Augustin et al. 2006). Существуют подобные коллекции данных для урду (Sagheer et al. 2009), фарси (Solimanpour et al. 2006), уйгурского (Ubul et al. 2013), арабского (Kharma et al. 1999; Al-Ohali et al. 2003; Mahmoud et al. 2014-03; Mezghani et al. 2012), испанского (Serrano et al. 2010), тамильского (Thadchanamoorthy et al. 2013), корейского (Dae-Hwan et al. 1996) и китайского (Jin et al. 2011; Liu et al. 2011), индонезийского (Aryan et al. 2011) и греческого (Kavallieratou et al. 2001). Перечисленные выше базы данных различаются как структурой, так и содержанием: некоторые

содержат изображения букв, другие предлагают изображения слов или целых предложений. Специальная база NIST (Wilkinson et al. 1992) и основанная на ней база MNIST (LeCun et al. 1998) содержат только изображения рукописных цифр. При этом создание этих баз данных происходит одинаково: участники заполняют заранее заготовленные бланки, которые затем сканируются в цифровой формат. Затем отсканированные изображения сопоставляются с текстом на компьютере, чтобы получились аннотированные изображения, готовые к экспериментам по распознаванию. Для русского языка существует база данных рукописных символов (Истратов & Федин 2013), которую мы будем использовать в экспериментах по распознаванию букв. Как следует из обзора баз данных для распознавания (Hussain et al. 2015-12-24), базы рукописных предложений и слов на русском языке не существовало до сих пор.

2.2. Общие подходы

Количество подходов к распознаванию рукописного текста достаточно обширно: нейронные сети, анализ графа, скрытые марковские модели, градиентный анализ, генетический алгоритм. Выбор конкретного подхода зависит от целевого языка или системы письменности.

С одной стороны, задача распознавания рукописного текста может расцениваться как задача классификации отдельных объектов. В этом случае используются такие методы обучения как метод опорных векторов (SVM), LVQ (learning vector quantization) или же различные типы нейронных сетей. Когда набор символов достаточно небольшой и они всегда четко отделены друг от друга, как например в случае распознавания рукописных цифр, успех распознавания может достигать почти 100% точности. Например, глубокие нейронные сети в работе (Ciresan et al. 2012) демонстрируют уровень ошибок 0.23% на наборе данных MNIST.

Другой подход к задаче основан на том, что рукописный текст это не набор случайных элементов, а (частично) упорядоченный набор объектов. Более того, форма каждого символа в курсивном тексте меняется в зависимости от расположенных рядом символов, из-за чего нейронным сетям гораздо сложнее угадать букву. В этом случае используются скрытые марковские модели (Toselli et al. 2004) и представления символов в виде графа (Кучуганов & Лапинская 2006). В работе Kala et al. 2010 анализ графа используется для преобразования изображения символа в граф, а затем используется генетический алгоритм для сравнения полученного графа с доступными шаблонами символов. Такая техника позволяет распознавать курсивный рукописный текст.

Скрытые марковские модели являются самым часто используемым методом в онлайн-распознавании рукописного текста. Данные при онлайн-распознавании хранятся в виде траектории пишущего инструмента, а траектория позволяет точно определить символ. Процент точности в системах онлайн-распознавания в недавнее время достиг почти 100% точности. Согласно работе Plötz & Fink 2009, возможно вычислить траекторию письма с отсканированного изображения, а затем использовать скрытую марковскую модель как в известных системах онлайн-распознавания. Авторы работы Zamora-Martínez et al. 2010 показывают, что лучшие результаты достигаются при комбинировании скрытых марковских моделей и ней-

ронных сетей.

3. Постановка задачи

Цель моей работы - создать OCR-программу для распознавания русского рукописного текста и базу данных с рукописными примерами для тренировки и тестирования программы.

Как было указано в предыдущем разделе, сейчас не существует стандартного набора данных с примерами русского рукописного текста. Что касается русского OCR, то большинство статей, ориентированных на распознавание рукописного текста, сосредоточены на обработке старославянского алфавита и автоматическое чтение исторических летописей (Зеленцов 2009). Системы, распознающие современный русский курсив, существуют, однако они являются коммерческими продуктами и не распространяются свободно. Кроме того, такие системы часто сосредоточены на рукопечатном, а не курсивном тексте. Например, система распознавания ABBYY OCR SDK способна читать машинные шрифты и рукопечатный текст, но не курсивный рукописный текст.

Заметим, что кроме проблемы распознавания текста со сканированного изображения существуют и другие классы задач, например, определение макета страницы (layout), которое включает определение количества колонок на странице, нахождение границ колонок, определение размера шрифта и выделение отдельных строк. Однако, в нашей работе мы остановимся только на задаче распознавания текста.

4. База русских рукописных предложений

4.1. Формирование анкеты

В нашей базе данных мы хотим видеть не только примеры предложений, написанных разными людьми, но и примеры отдельных букв, написанных теми же людьми. Кроме того, желательно иметь информацию о пишущих, чтобы возможно было различать их между собой (например, для задач установления авторства). Учитывая эти требования, мы сформировали анкету для сбора рукописных данных.

Мы разрабатывали анкету с учетом опыта таких проектов как NIST (Wilkinson et al. 1992), IAM (Marti & Bunke 2002) и KHATT (Mahmoud et al. 2014-03), однако общий вид (дизайн, макет и состав) анкеты является нашей оригинальной разработкой.

Анкета состоит из четырех частей:

1. Информация о заполняющем: имя, родной город, возраст, пол. Также в этом блоке печатается номер заполняемой анкеты.
2. Написание букв: в этом блоке заполняющий должен написать все буквы русского алфавита своим обычным курсивным почерком через пробел. Сначала пишутся только

строчные буквы, затем – только заглавные. Буквы пишутся не в алфавитном порядке, а в случайном, заранее заданном в бланке.

Порядок строчных букв:

н э м з р о к т у й ю ъ ё ф в ц б л е и ы д х щ ь п а ж г ч ш с я

Порядок заглавных букв:

Ж Ч Е Р С Б Ф Й А Н П Г Щ Я К Ю З У В Л Ш Ц О Т Ё М Ъ Д Э Ы И Ь Х

Объясним, почему мы приняли решение печатать буквы в случайном порядке. В тестовой версии анкеты буквы располагались в алфавитном порядке, и заметив это, респонденты заполняли блок с буквами по памяти. Однако, очень часто в этом случае респонденты пропускали какую-либо букву алфавита, в особенности "ё" или "й" или путали местами мягкий и твердый знаки. Поэтому было принято решение изменить порядок букв, чтобы респондент более внимательно заполнял анкету.

3. Написание слов: в этом блоке пишущему предложено переписать от руки от двух до четырех слов. Слова для этого блока извлекаются из частотного списка униграмм[1], представленных на сайте НКРЯ[2]. Из этого списка слов мы извлекаем 62 слова с наибольшим количеством частотных пар букв (см. Приложение А), а затем случайным образом выбираем по несколько слов для каждой анкеты так, чтобы общая длина слов не выходила за границы листа А4. Таким образом, наборы слов получаются разные в разных анкетах.

4. Написание предложений: в этом блоке необходимо написать от руки три фрагмента текста. Первые два фрагмента одинаковы для всех анкет: панграмма[3] "Широкая электрификация южных губерний даст мощный толчок подъёму сельского хозяйства." и цитата из Конституции РФ "Человек, его права и свободы являются высшей ценностью. Признание, соблюдение и защита прав и свобод человека и гражданина — обязанность государства.". Третий фрагмент длиной от одного до трех предложений выбирается случайным образом из НКРЯ.

Шаблон анкеты написан в системе LaTex. Для каждого варианта анкеты в этот шаблон автоматически вставляются меняющиеся части – номер анкеты, набор слов и третий фрагмент текста. Затем каждая анкета компилируется в файл PDF и распечатывается. При компиляции также генерируется лог-файл, в котором записывается номер анкеты, набор слов и третий фрагмент текста. Это необходимо для того, чтобы после сканирования возможно было автоматически сопоставить изображение с соответствующим ему текстом. Всего было сгенерировано 1082 варианта анкеты. Бланк анкеты печатается красными чернилами; пример пустого бланка представлен на рисунке 1.

Некоторые предложения из НКРЯ включают в себя слова, написанные латиницей, и цифры, так что база подходит и для задач определения системы письменности. Поскольку каждый респондент имеет собственный индивидуальный номер, то также можно использовать базу и для определения авторства.

[1] Частоты n-грамм были скачаны по ссылке: `www.ruscorpora.ru/corpora-freq.html`.

[2] Национальный корпус русского языка доступен по ссылке `http://www.ruscorpora.ru`.

[3] Панграмма – короткий текст, использующий все или почти все буквы алфавита.

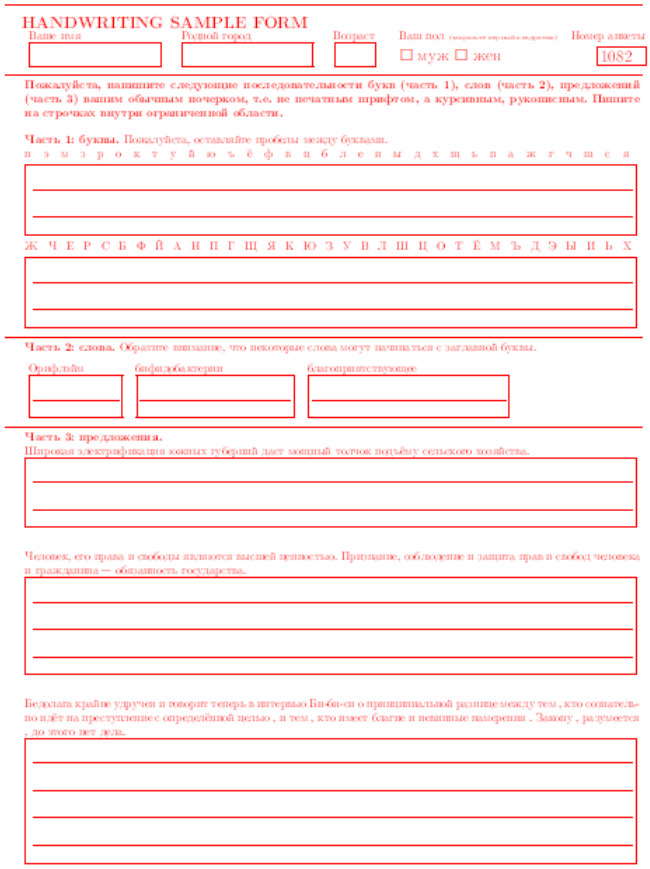


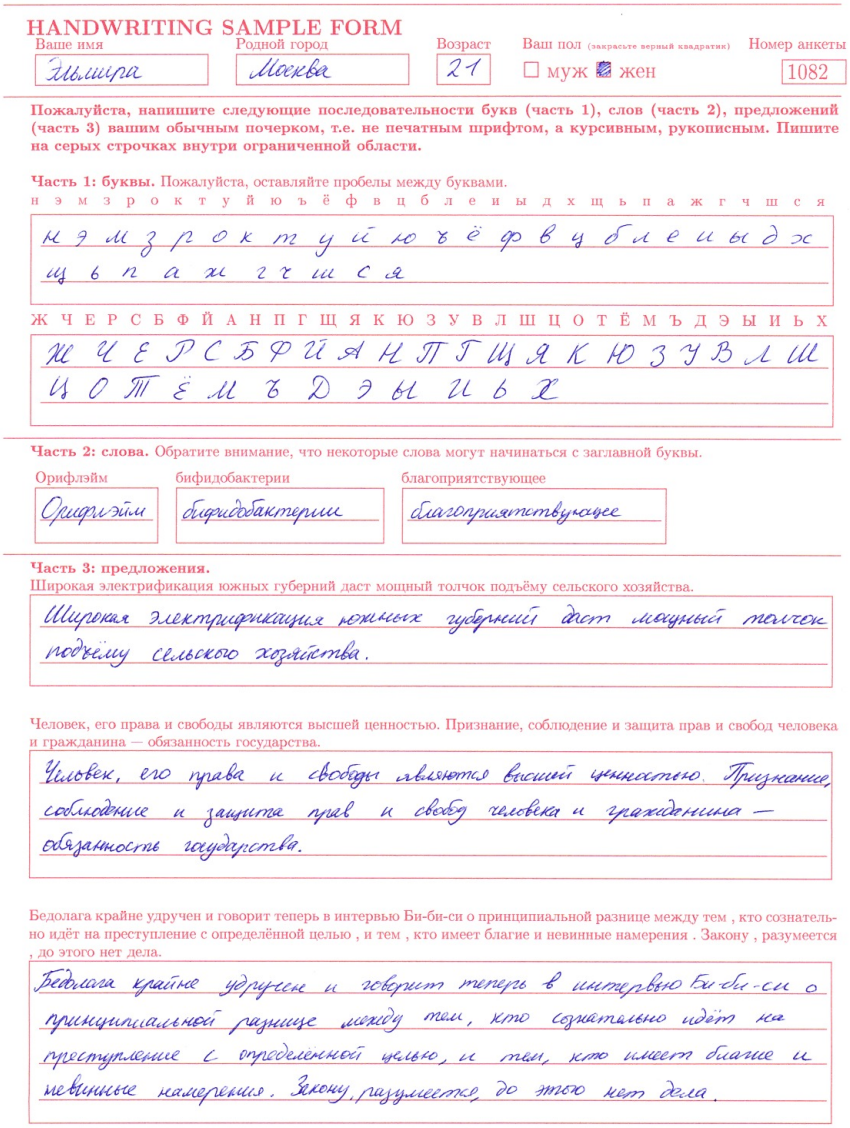
Рис. 1. Пример пустой анкеты

Анкеты были заполнены около 300 респондентами, каждый респондент заполнил от 1 до 5 бланков. Пример заполненного бланка представлен на рисунке 2.

4.2. Обработка бланков

Важным этапом создания базы данных является обработка бланков (см. рис. 3). Обработка осуществляется в два этапа. На первом этапе вручную составляется таблица с метаданными по каждому бланку: в таблицу записывается номер бланка, имя, родной город и возраст заполняющего, индивидуальный номер заполняющего, а также пищущий инструмент (гелевая или шариковая ручка) и цвет чернил (черный или синий). Заполненные бланки сканируются в цвете с разрешением 150 dpi (dots per inch) и сохраняются в формате PNG. Файлу с рисунком присваивается название вида “номер анкеты_номер заполняющего.png”, например, “1082_1.png”.

Второй этап обработки проходит автоматически с помощью модулей `scikit-image` (van der Walt et al. 2014) и `OpenCV` (Bradski) для Python . По положению горизонтальной черты в начале бланка определяется угол наклона бумаги: если угол не равен 0, то наклон исправляется. Затем осуществляется разделение цветного изображения на два бинарных изображения за счет различий цветовой палитры заполненного бланка: форма печатается красными чернилами, а респонденты заполняют анкету синими или черными чернилами (см. рис. 4).



HANDWRITING SAMPLE FORM

Ваше имя: Эльмира | Родной город: Москва | Возраст: 21 | Ваш пол (закрасьте верный квадратик): ☐ муж ▨ жен | Номер анкеты: 1082

Пожалуйста, напишите следующие последовательности букв (часть 1), слов (часть 2), предложений (часть 3) вашим обычным почерком, т.е. не печатным шрифтом, а курсивным, рукописным. Пишите на серых строчках внутри ограниченной области.

Часть 1: буквы. Пожалуйста, оставляйте пробелы между буквами.

н э м з р о к т у й ю ъ ё ф в ц б л е и ы д х щ ь п а ж г ч ш с я

Ж Ч Е Р С Б Ф Й А Н П Г Щ Я К Ю З У В Л Ш Ц О Т Ё М Ъ Д Э Ы И Ь Х

Часть 2: слова. Обратите внимание, что некоторые слова могут начинаться с заглавной буквы.

Орифлэйм | бифидобактерии | благоприятствующее

Часть 3: предложения.

Широкая электрификация южных губерний даст мощный толчок подъёму сельского хозяйства.

Человек, его права и свободы являются высшей ценностью. Признание, соблюдение и защита прав и свобод человека и гражданина — обязанность государства.

Бедолага крайне удручен и говорит теперь в интервью Би-би-си о принципиальной разнице между тем , кто сознательно идёт на преступление с определённой целью , и тем , кто имеет благие и невинные намерения . Закону , разумеется , до этого нет дела.

Рис. 2. Пример заполненной анкеты

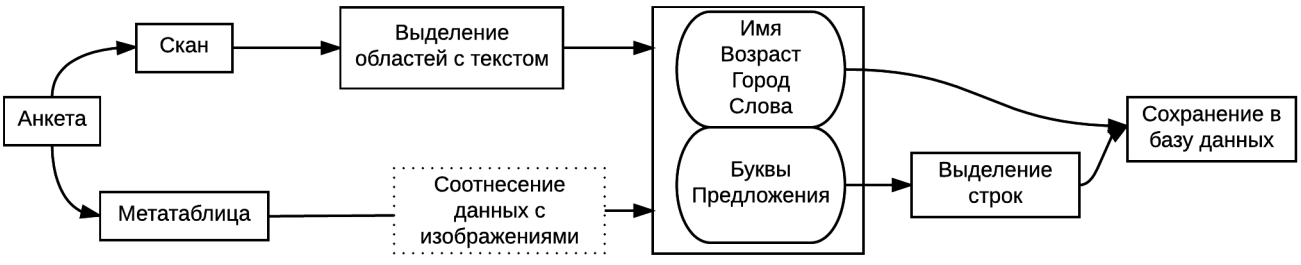


Рис. 3. Процесс обработки бланков

Затем из бинаризованного изображения полей анкеты вырезаются области с именем,

городом, возрастом, строчные буквы, прописные буквы, строка со словами и три области с предложениями. Так как все бланки одинаковые, то области вырезания задаются заранее. Затем в каждой области обрезаются лишние поля и выделяются строки.

Определение границ строк происходит следующим образом. Сначала по изображению фрагмента строится гистограмма: каждой строке пикселей сопоставляется количество черных пикселей в этой строке. По этой гистограмме строится полиномиальная функция, график которой приближен к значениям гистограммы. Затем осуществляется поиск минимумов этой функции на промежутке от 0 до значения, равного высоте изображения. Минимумы – это как раз те строки пикселей, в которых проходит граница строки. В идеальном случае количество черных пикселей на этой строке равно нулю, однако в большинстве случаев этого не происходит, так как выносные верхние и нижние элементы букв часто пересекают эту границу. Пример выделения строк представлен на рисунке 5: по вырезанному из сканированной анкеты предложению (верхнее изображение) строится график, на котором определяются минимумы (выделены зелеными точками). На рисунке 6 показан пример ситуации, когда некоторые выносные элементы могут обрезаться.

Затем строки делятся на слова или буквы. Каждое выделенное изображение сохраняется в формате png. Таким образом мы сохраняем области с буквами и текстовыми фрагментами, не поделенные на строки, затем – отдельные слова и буквы.

В итоге обработки сканированных анкет мы получили несколько типов аннотированных изображений: рукописные буквы русского языка, слова на русском языке, рукописные числа, целые предложения.

4.3. Онлайн база-данных

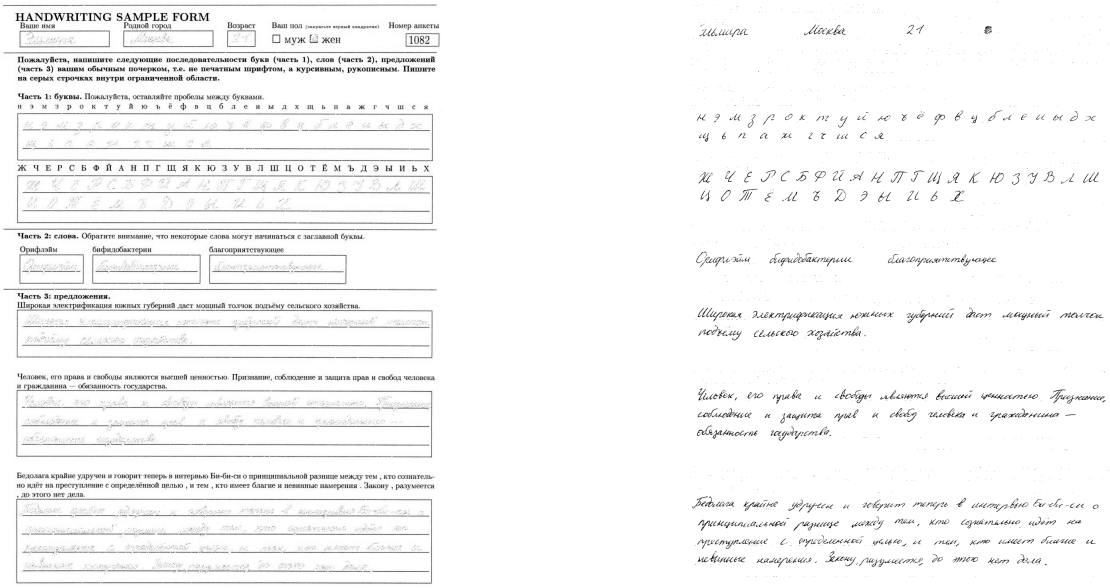
Созданная нами база данных доступна онлайн (`www.elmiram.github.io/ruhwr/`). Онлайн база данных позволяет исследователю скачать только те материалы, которые ему необходимы: возможно скачать только изображения букв, чисел, слов или предложений. Кроме того на сайте доступен архив со всеми пустыми бланками в формате pdf и со всеми сканами заполненных бланков в формате png.

База содержит 19932 изображений отдельных букв, 21700 изображений слов, 302 чисел, 906 предложений.

Было заполнено 303 анкеты 282 респондентами. Возраст респондентов от 16 до 77 лет, 2/3 респондентов женского пола, и лишь треть – мужского.

5. Распознавание рукописного текста

В своей работе мы хотим испробовать несколько разных методов распознавания с использованием двух баз данных: базы данных символов (Истратов & Федин 2013) и нашей базы данных (`http://elmiram.github.io/ruhwr/`), описанной в предыдущем разделе. Мы



(a) Анкета

(b) Поля анкеты

Рис. 4. Бинаризованные изображения

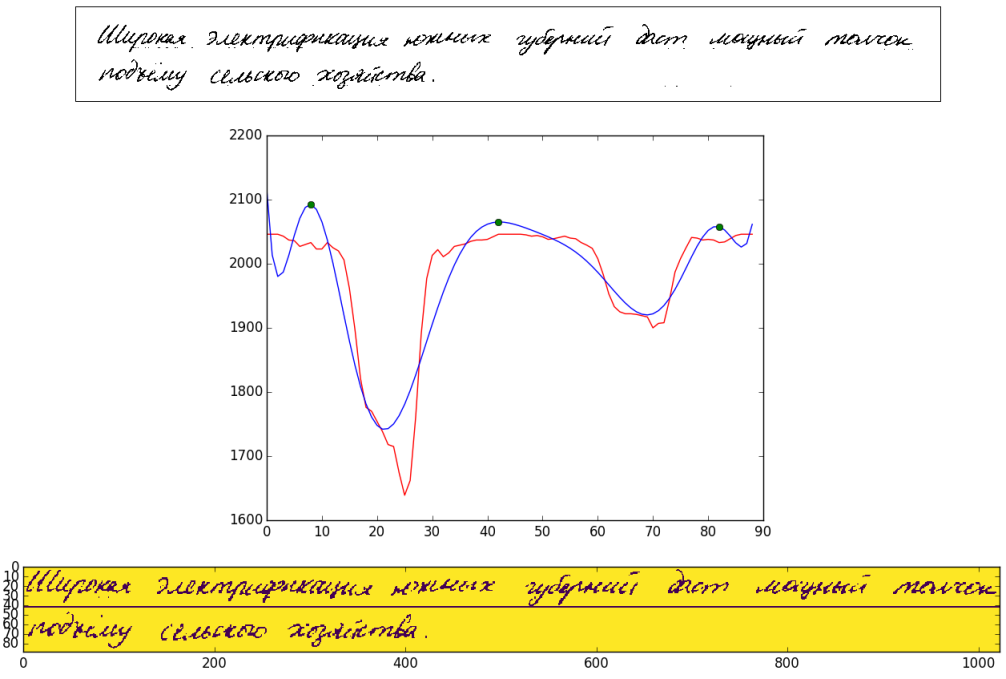


Рис. 5. Выделение строк. Символы не пересекают границу строки.

используем методы, уже показавшие себя в задаче распознавания рукописного текста: метод опорных векторов и нейронные сети. Этапы процесса распознавания текста по изображению представлены на рисунке 7.

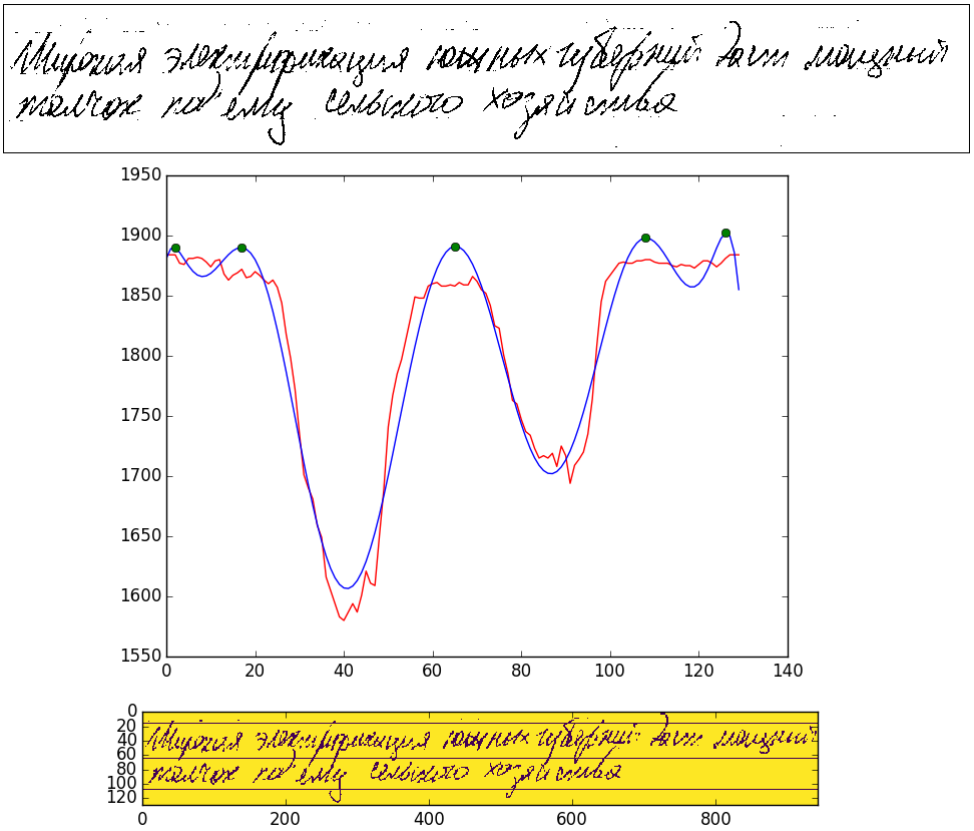


Рис. 6. Выделение строк. Выносные элементы символов пересекают границу строки.

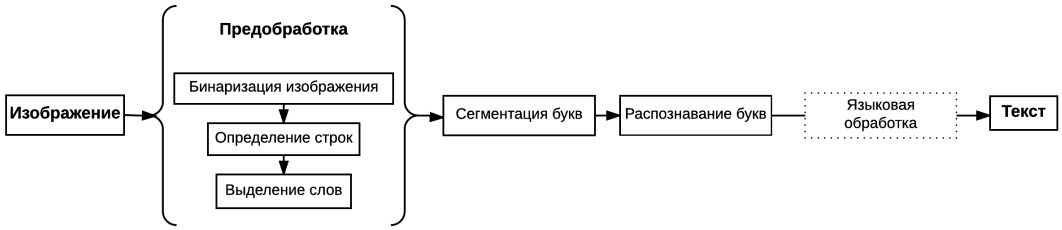


Рис. 7. Процесс распознавания

5.1. Распознавание букв

5.1.1. Классификация методом опорных векторов

Задача классификации состоит в определении, к какому классу из двух или более изначально известных относится данный объект. Обычно таким объектом является вектор в n-мерном вещественном пространстве . В нашем случае размерность вектора равна количеству пикселей в изображении (400 пикселей), и координаты вектора получают значение 1 или 0, в зависимости от того закрашен, ли рассматриваемый пиксель. Необходимо раскласифицировать изображения в один из 74 классов. Для каждого класса имеются образцы — объекты, про которые заранее известно, к какому классу они принадлежат.

В методе опорных векторов каждый объект представляется в виде точки в пространстве размерности p, и задача метода – разделить данные точки гиперплоскостью размерности (p-1). Данные могут быть разделены с помощью различных гиперплоскостей, но лучшая гиперплоскость – гиперплоскость, при построении которой разделение и разница между классами максимальна. Например, в случае с точками на плоскости мы выберем прямую, максимально далеко проходящую от точек, таким образом, что расстояние от нее до ближайшей точки с каждой стороны будет максимальным.

Для обучения мы используем метод опорных векторов, реализованный в модуле `scikit-learn` для Python (Pedregosa et al. 2011). Мы провели три эксперимента: 1) сначала обучали метод опорных векторов только на цифрах, чтобы сравнить результат метода с данными MNIST, 2) затем только на строчных буквах, 3) затем на всех символах из базы символов Истратов & Федин 2013. Результат работы представлен в таблице 1. Можно заметить, что чем меньше набор данных используется при классификации, тем лучше результат тестирования.

Таблица 1. Метод опорных векторов

набор данных	точность	полнота	f-мера	количество примеров
цифры	0.92	0.92	0.92	483
строчные буквы	0.83	0.83	0.83	8501
все символы	0.70	0.70	0.69	15287

5.1.2. Нейронная сеть

Нейронная сеть – это компьютерная модель нейронов, соединенных между собой, имитирующая работу нейронов головного мозга. В искусственной нейронной сети каждый нейрон – это небольшая функция, которая получает на вход несколько значений и возвращает некоторый результат в зависимости от этих значений. Работа нейронной сети состоит в преобразовании входного вектора в выходной вектор, причем это преобразование задается весами нейронной сети. Задачу распознавания рукописных букв по изображению 20x20 пикселей можно переформулировать для нейронной сети следующим образом: необходимо построить нейронную сеть с 400 входами для входного вектора из 400 двоичных символов и 32 выходами, которые помечены буквами. Если на входе нейронной сети изображение буквы "A", то максимальное значение выходного сигнала достигается на выходе "A". Аналогично нейронная сеть работает для всех 32 букв.

При построении нейронной сети необходимо выбрать ее архитектуру (тип нейронов, количество слоев нейронов, количество нейронов в каждом слое) и подобрать веса для этой сети. Для построения и обучения нейронной сети мы использовали глубокое обучение (Deep Learning), реализованное в модуле `nolearn` для Python.

Для распознавания символов русского алфавита мы провели серию экспериментов с использованием разных архитектур нейронной сети и разных наборов данных. Типы архитектур и результаты экспериментов представлены в таблице 2. Мы использовали нейронные

сети с одним скрытым слоем, то есть в колонке “архитектура” в таблице ниже указаны через дефис количество нейронов во входном слое, скрытом и выходном слое соответственно.

Таблица 2. Нейронная сеть

архитектура	набор данных	точность	полнота	f-мера	количество примеров
400-300-74	все символы	0.75	0.75	0.75	33632
400-800-74	все символы	0.75	0.75	0.75	33632
400-300-32	строчные буквы	0.90	0.90	0.90	18702
400-300-10	цифры	0.93	0.93	0.93	1063
400-2000-10	цифры	0.94	0.94	0.94	1063

5.2. Распознавание слов

Для распознавания слов мы используем натренированные модели, описанные в предыдущих подразделах. В экспериментах по распознаванию мы используем рукописные слова из нашей базы данных. Мы ограничили список слов только именами респондентов и названиями городов, чтобы сократить объем распознаваемого словаря на этом этапе.

Перед непосредственным применением моделей необходимо выровнять наклон слова по отношению к горизонтальной строке. Для этого мы строим по изображению слова “скелет” этого слова - то есть аналогичное изображение, в котором ширина линии не превышает одного пикселя. Затем по такому скелету находим все прямые линии с помощью преобразования Хафа (Galamhos et al. 1999, Duda & Hart 1972). Затем мы выбираем из полученных объектов все линии, угол наклона которых между -60 и 60, и считаем по ним средний угол наклона. Полученное значение мы считаем углом наклона почерка и в соответствии с ним выпрямляем изображение. Последовательно все этапы показаны на рисунке 8 на примерах двух почерков с разным наклоном: A - оригинальное изображение, B - “скелет” слова, C - прямые линии, построенные по скелету, D - изображение с исправленным наклоном.

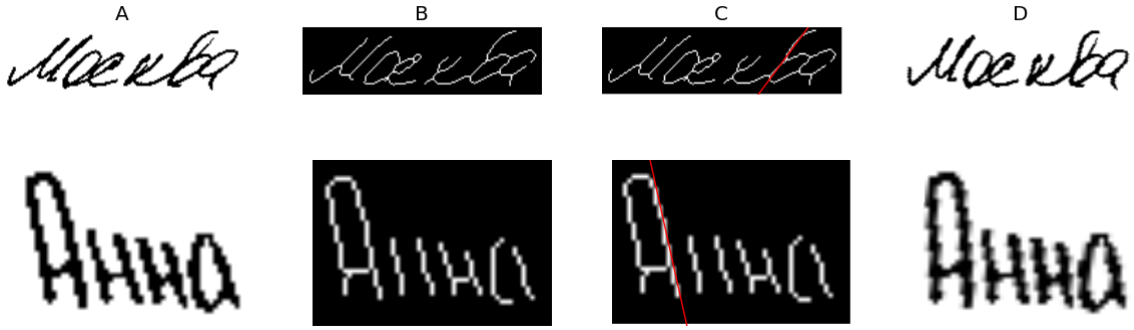


Рис. 8. Исправление наклона почерка

Следующий этап на пути к распознаванию текста – это разделение слова на символы. В связи с этой задачей в Sayre 1973 был сформулирован “парадокс Сайра”: чтобы распознать

букву, необходимо знать, где она начинается и заканчивается, но чтобы знать, где она начинается и заканчивается, сначала ее нужно распознать. Существуют подходы к преодолению этого парадокса. Например, в Timár et al. 2003 предлагаются два пути: либо отказаться от распознавания отдельных символов и перейти к распознаванию целых слов, что приемлемо в случае с небольшим набором возможных слов (например, при распознавании банковских чеков), либо делить слова не на символы, а на минимальные черты и дуги (примитивные сегменты), из которых затем строится буква и передается в распознающую модель. Мы решили использовать более простой и грубый метод: последовательно выделять по 10 пикселей изображения, пока модель не распознает букву с уверенностью, выше определенного порога, например, 70%. Как только модель достаточно уверенно распознала символ, переходим к поиску нового символа подобным же образом. Основная проблема этого метода состоит в том, что нам неизвестно, сколько символов в рассматриваемом слове, и поэтому алгоритм может выделить больше или меньше символов. Эта проблема решается для некоторых почерков другим, еще более простым методом: если респондент при письме не соединяет буквы, то отделить символы друг от друга не составляет труда, как например, в слове “Анна” на рисунке 8. Затем к каждому выделенному символу мы применяем поочередно натренированные модели и соединяем их предсказания в слово.

Однако работа подобного алгоритма не оправдала наши ожидания: в контексте слова верно определяются лишь 30% символов, тогда как изолированные буквы распознаются с точностью около 90%. В таблице 3 приведены примеры результатов обработки слова “Москва” с исправленным наклоном (см. рис. 8). Цветом выделены те символы, которые распознались верно. Чаще всего в этом слове верно были определены буквы “М” и “в”.

Таблица 3. Распознавание символов в рукописном тексте

модель	М	о	с	к	в	а
Нейронная сеть, 400-300-74, все символы	М	э	е	ж	В	е
Метод опорных векторов, все символы	ж	Ш	В	В	В	ж
Нейронная сеть, 400-300-32, только строчные	м	м	я	к	в	х
Нейронная сеть, 400-800-74, все символы	М	Ш	е	ы	В	т

В следующем шаге мы добавили к распознаванию словарь, состоящий из имен и названий городов, встретившихся в собранных анкетах. После распознавания символов с помощью моделей мы ищем в словаре близкие к выдаче модели слова по расстоянию Левенштейна. В этом случае, результат стал лучше: слова распознаются всегда, если на этапе выделения символов было выделено верное количество символов.

Обученные модели показывают бедные результаты, потому что при слитном написании буквы значительно изменяют свою форму в зависимости от соседних символов. Кроме того, отрицательно на результате сказывается необходимость преобразовывать каждый выделенный в слове символ в изображение 20x20 пикселей, так как на вход нашим моделям необходимо подавать вектор с 400 координатами. При сжатии изображения символа возможна потеря

значительной части важной информации. Также на качество влияет то, что некоторые буквы при написании похожи друг на друга: например, “е” и “с”, как можно заметить из таблицы 3, заглавные и строчные символы одной и той же буквы, например, “в” и “В” или “с” и “С”, а также “Ш”, “ш”, “Щ” и “Щ”.

5.3. Результаты в других работах

Базовой линией при распознавании символов считается мера средней темноты картинки. Для каждого символа считается среднее отношение темных пикселей к светлым, а новые изображения классифицируются по близости их меры темноты к какому-либо из известных классов. При распознавании цифр эта базовая линия находится на уровне 20-25% точности, при распознавании букв алфавита – на уровне 7-10%. В задаче распознавания отдельных символов мы далеко продвинулись за эту базовую отметку, однако в задаче распознавания букв внутри слова мы лишь на 20% превзошли минимальный порог.

Что касается результатов распознавания слов в других языках, то для английского языка процент успеха колеблется в области 80–90% (Zamora-Martínez et al. 2014). Этот результат меньше чем например для французского языка: на наборе данных RIMES была достигнута точность 94.87% (Grosicki & El-Abed 2011). Разница в качестве объясняется набором лексикона используемого при обучении: в RIMES всего 1600 уникальных слов, тогда как в IAM более 10000 слов. Высокие показатели распознавания слов (93.37%) были показаны на арабской базе IFN/ENIT (El Abed & Märgner 2011). Во всех этих соревнованиях кроме простого распознавания символов использовался словарь с заданными словами.

6. Заключение

Распознавание рукописного текста – одна из самых исследуемых и пока что наименее поддающихся решению задач в области компьютерного зрения. Для задачи распознавания рукописного текста стандартный набор данных большого объема является необходимостью.

После сбора и обработки данных мы создали веб-ресурс для распознавания русского рукописного текста. Обработанные данные доступны академическому сообществу в форме онлайн базы данных. Такая база данных позволяет другим исследователям пропустить этап сбора данных и тестировать свои распознающие системы на уже подготовленном наборе данных. Создание единого набора данных позволяет сделать результаты разных распознающих систем более сравнимыми между собой. Включение в базу изображений цифр, букв, слов и предложений позволяет использовать ее для тестирования систем OCR (Optical Character Recognition), IWR (Intelligent Word Recognition) и ICR (Intelligent Character Recognition).

Также результатом нашей работы стал набор моделей для распознавания русского рукописного текста. Успех моделей при распознавании индивидуальных символов достигает 94%, а при распознавании слов пока что лишь 30%.

В дальнейшей работе необходимо опробовать на нашей базе данных методы IWR (Intelligent

Word Recognition): распознавание слов не по буквам, а целиком. Также нам кажется интересной возможность применить в будущем к нашему набору данных методы онлайн-распознавания: восстановить траекторию пишущего инструмента по изображению и использовать для распознавания уже существующие модели, встроенные в современные сенсорные устройства ввода.

A Приложение. Список слов для анкеты.

Ниже представлен список слов в алфавитном порядке:
административно-территориальными
Азербайджанфильм
безвременье
бифидобактерии
благоприятствующее
Бомбоубежище
взаимодополняющими
видеоизображение
влагосвязывающая
Выдвижение
Герцогиню
Главпочтамта
глубочайшаго
ГОСУДАРСТВЕННОГО
гражданско-процессуальных
десантно-высадочные
жизнеобеспечивающих
злодеяниях
информационно-телекоммуникационные
Кайзер-Вильгельм-Института
Ключом
Крепко-накрепко
ЛАБОРАТОРНАЯ
Медико-хирургическую
МЕТРОПОЛИТЕН
Находящимся
Неиспользуемые
Нижневолжского
Нотариусов
обществоведческой
Объявляйте
окислительно-восстановительного
Орифлэйм
Остаётся
ответственнейшей
Откашлялся
Пасху
Переглядываясь
посрамлённый
пошучу

поэтизируя
предпосылкою
ПРЕДПРИНИМАТЕЛЬСТВУ
Приплюснутый
профессорско-преподавательской
псевдореминисценции
распространяется
разбрызгивать
размножаются
Раскрасневшийся
Рейхсфюреру
Сверхмощный
Скорее
СОЦИАЛ-ДЕМОКРАТИЧЕСКАЯ
Спортплощадка
Тензодатчики
угнетению
умудряешься
Фотосъемка
Ханьчжоу
хлещущую
Шарикоподшипниковый

Список литературы

А. В. Кучуганов & Г. В. Лапинская. Распознавание рукописных текстов. In *АВ Кучуганов, ГВ Лапинская–Материалы международной научной конференции Ижевск*, pages 13–17, 2006.

А. Ю. Истратов & Н. А. Федин. Формирование базы сегментированных рукописных символов русского алфавита. pages 36–41, 2013.

И. А. Зеленцов. Метод распознавания древнерусской скорописи. pages 116–131, 2009.

Yousef Al-Ohali, Mohamed Cheriet, & Ching Suen. Databases for recognition of handwritten arabic cheques. 36(1):111–121, 2003. URL `http://www.sciencedirect.com/science/article/pii/S003132030200064X`.

Peb Ruswono Aryan, Iping Supriana, & Ayu Purwarianti. Development of indonesian handwritten text database for offline character recognition. In *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*, pages 1–4. IEEE, 2011. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6021582`.

Emmanuel Augustin, Matthieu Carré, Emmanuèle Grosicki, Jean-Marie Brodin, Edouard Geoffrois, & Françoise Prêteux. Rimes evaluation campaign for handwritten mail processing. In *Proceedings of the Workshop on Frontiers in Handwriting Recognition*, volume 1, 2006. URL `http://www-artemis.it-sudparis.eu/Publications/library/Augustin-IWFHR2006.pdf`.

G. Bradski. *Dr. Dobb's Journal of Software Tools*.

Dan Ciresan, Ueli Meier, & Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6248110`.

K. I. M. Dae-Hwan, Young-Sup Hwang, PARK Sang-Tae, K. I. M. Eun-Jung, PAEK Sang-Hoon, & BANG Sung-Yang. Handwritten korean character image database PE92. 79(7):943–950, 1996. URL `http://search.ieice.org/bin/summary.php?id=e79-d_7_943`.

Richard O Duda & Peter E Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

Haikal El Abed & Volker Märgner. Icdar 2009-arabic handwriting recognition competition. *International Journal on Document Analysis and Recognition (IJDAR)*, 14(1):3–13, 2011.

C Galamhos, Jose Matas, & Josef Kittler. Progressive probabilistic hough transform for line detection. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1. IEEE, 1999.

Emmanuele Grosicki & Haikal El-Abed. Icdar 2011-french handwriting recognition competition. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1459–1463. IEEE, 2011.

Jonathan J. Hull. A database for handwritten text recognition research. 16(5):550–554, 1994. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=291440`.

Raashid Hussain, Ahsen Raza, Imran Siddiqi, Khurram Khurshid, & Chawki Djeddi. A comprehensive survey of handwritten document benchmarks: structure, usage and evaluation. 2015(1):1–24, 2015-12-24. ISSN 1687-5281. doi: 10.1186/s13640-015-0102-5. URL http://link.springer.com/article/10.1186/s13640-015-0102-5.

Lianwen Jin, Yan Gao, Gang Liu, Yunyang Li, & Kai Ding. SCUT-COUCH2009—a comprehensive online unconstrained chinese handwriting database and benchmark evaluation. 14(1):53–64, 2011. URL http://link.springer.com/article/10.1007/s10032-010-0116-6.

Rahul Kala, Harsh Vazirani, Anupam Shukla, & Ritu Tiwari. Offline handwriting recognition using genetic algorithm. 2010.

Ergina Kavallieratou, Nikos Liolios, E. Koutsogeorgos, Nikos Fakotakis, & G. Kokkinakis. The GRUHD database of greek unconstrained handwriting. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 561–565. IEEE, 2001. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=953852.

Nawwaf Kharma, Maher Ahmed, & Rabab Ward. A new comprehensive database of handwritten arabic words, numbers, and signatures used for OCR testing. In *Electrical and Computer Engineering, 1999 IEEE Canadian Conference on*, volume 2, pages 766–768. IEEE, 1999. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=808042.

Yann LeCun, Léon Bottou, Yoshua Bengio, & Patrick Haffner. Gradient-based learning applied to document recognition. 86(11):2278–2324, 1998. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=726791.

Cheng-Lin Liu, Fei Yin, Da-Han Wang, & Qiu-Feng Wang. CASIA online and offline chinese handwriting databases. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 37–41. IEEE, 2011. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6065272.

Sabri A. Mahmoud, Irfan Ahmad, Wasfi G. Al-Khatib, Mohammad Alshayeb, Mohammad Tanvir Parvez, Volker Märgner, & Gernot A. Fink. KHATT: An open arabic offline handwritten text database. 47(3):1096–1112, 2014-03. ISSN 0031-3203. doi: 10.1016/j.patcog.2013.08.009. URL http://www.sciencedirect.com/science/article/pii/S0031320313003300.

U.-V. Marti & Horst Bunke. The IAM-database: an english sentence database for offline handwriting recognition. 5(1):39–46, 2002.

Amine Mezghani, Slim Kanoun, Mahdi Khemakhem, & Haikal El Abed. A database for arabic handwritten text image recognition and writer identification. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 399–402. IEEE, 2012. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6424426.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, & E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Thomas Plötz & Gernot A. Fink. Markov models for offline handwriting recognition: a survey. 12 (4):269–298, 2009.

Malik Waqas Sagheer, Chun Lei He, Nicola Nobile, & Ching Y. Suen. A new large urdu database for off-line handwriting recognition. In *Image Analysis and Processing–ICIAP 2009*, pages 538–546. Springer, 2009. URL http://link.springer.com/chapter/10.1007/978-3-642-04146-4_58.

Kenneth M Sayre. Machine recognition of handwritten words: A project report. *Pattern recognition*, 5(3):213–228, 1973.

Nicolás Serrano, Francisco Castro, & Alfons Juan. The RODRIGO database. In *LREC*, 2010. URL http://lexitron.nectec.or.th/public/LREC-2010_Malta/pdf/477_Paper.pdf.

Farshid Solimanpour, Javad Sadri, & Ching Y. Suen. Standard databases for recognition of handwritten digits, numerical strings, legal amounts, letters and dates in farsi language. In *Tenth International workshop on Frontiers in handwriting recognition*. Suvisoft, 2006. URL https://hal.archives-ouvertes.fr/inria-00103983/.

S. Thadchanamoorthy, N. D. Kodikara, H. L. Premaretne, Umapada Pal, & Fumitaka Kimura. Tamil handwritten city name database development and recognition for postal automation. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 793–797. IEEE, 2013. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6628727.

Gergely Timár, Kristóf Karacs, & Csaba Rekeczky. Analogic preprocessing and segmentation algorithms for offline handwriting recognition. 12(6):783–804, 2003.

Alejandro Hector Toselli, Alfons Juan, Jorge González, Ismael Salvador, Enrique Vidal, Francisco Casacuberta, Daniel Keysers, & Hermann Ney. Integrated handwriting recognition and interpretation using finite-state models. 18(4):519–539, 2004. URL http://www.worldscientific.com/doi/abs/10.1142/S0218001404003344.

Kurban Ubul, Mavjuda Zunun, Alim Aysa, Nurbiya Yadikar, & Umut Yunus. Creation of uyghur offline handwritten database. In *Systems, Signal Processing and their Applications (WoSSPA), 2013 8th International Workshop on*, pages 291–295. IEEE, 2013.

Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, & the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014. ISSN 2167-8359. doi: 10.7717/peerj.453. URL http://dx.doi.org/10.7717/peerj.453.

R. Allen Wilkinson, Jon Geist, Stanley Janet, PATRICKJ Grother, Christopher JC Burges, Robert Creecy, Bob Hammond, Jonathan J. Hull, N. J. Larsen, Thomas P. Vogl, & others. *The first census optical character recognition system conference*, volume 184. US Department of Commerce, National Institute of Standards and Technology, 1992. URL http://www.nist.gov/manuscript-publication-search.cfm?pub_id=905664.

Francisco Zamora-Martínez, Volkmar Frinken, S Espana-Boquera, Maria José Castro-Bleda, Andreas Fischer, & Horst Bunke. Neural network language models for off-line handwriting recognition. *Pattern Recognition*, 47(4):1642–1652, 2014.

Francisco Zamora-Martínez, M. J. Castro-Bleda, S. España-Boquera, & Jorge Gorbe-Moya. Unconstrained offline handwriting recognition using connectionist character n-grams. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–7. IEEE, 2010.