Федеральное государственное автономное образовательное учреждение

высшего образования

«Национальный исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук

Образовательная программа Прикладная математика и информатика

бакалавриат

**01.03.02 Прикладная математика и информатика**

**О Т Ч Е Т**

**по учебной практике**

Выполнил студент гр.БПМИ-177

Семененя Яна Игоревна

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*(подпись)*

**Проверили:**

ИП, Костюков Михаил Валериевич

*\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(оценка) (подпись)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

**2017/2018 уч.г.**

Оглавление

[**Введение**](#_1fob9te) **3**

[**Основная часть**](#_3znysh7) **4**

[Создание анкеты](#_g9v0bql8u0jf) 4

[Сканирование анкет и их обработка](#_8gcf4kfmy4g0) 7

[Извлечение букв](#_yj6nv8kv0i01) 7

[Формирование датасета](#_n4w2mt1iow5c) 9

[Обучение алгоритмов ML](#_yt8hg380x225) 9

[**Заключение**](#_2et92p0) **10**

## Введение

Проблема распознания рукописного текста, несмотря на обилие коммерческих решений, не перестаёт быть актуальной. Для её решения необходимы качественные датасеты рукописных символов. К сожалению, наиболее полные собрания образцов не допускают коммерческого использования или находятся в закрытом доступе. Также открытой является проблема отсутствия качественных датасетов для малораспространённых видов письменности, а также узкоспециализированных символов.

Целью настоящей работы являлась разработка автоматизации процесса и ПО для сбора датасетов рукописных символов, применимого для широкого круга практических задач, и способного облегчить работу исследователя на всех этапах – от первоначального создания анкет до последующей их обработки для придания вида, максимально удобного для применения в обучении моделей.

Руководителем практики были поставлены следующие задачи:

1. Моделирование анкеты для сбора данных

2. Создание выборки рукопечатного английского текста из более чем 10 различных почерков

3. Обработка собранных данных

4. Использование полученной выборки для алгоритмов машинного обучения

## Основная часть

Традиционный подход к созданию датасетов рукописных букв включает в себя следующие этапы:

* отрисовка анкеты-шаблона, копии которой будут розданы респондентам для заполнения и написания букв
* сканирование заполненных анкет и последующее улучшение их качества программными методами (устранение шума, исправление баланса белого)
* подготовка к извлечению букв и собственно последующее извлечение
* сохранение полученного набора букв и виде датасета, представленного в удобном для использования в приложениях виде

Описанный выше подход является общепринятым, однако техническая сторона выполнения вышеуказанных пунктов может меняться в широких пределах. Во время предварительного обзора материалов, однако, не было найдено библиотек, находящихся в открытом доступе и позволяющих проделать всю последовательность действий без необходимости обращения к низкоуровневым командам. Таким образом, было выдвинуто и одобрено предложение создать подобное решение.

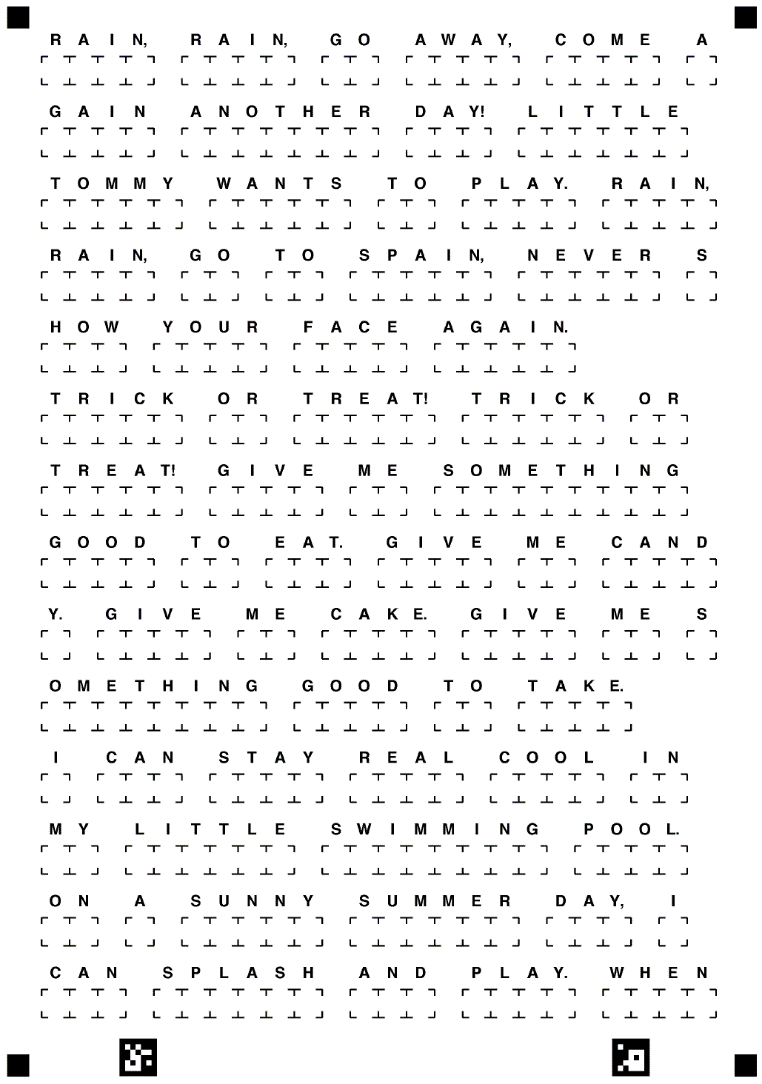
В данной работе в качестве языка разработки был выбран Python, как один из наиболее удобных языков программирования как для прототипирования, так и для продакшн-решения. Для работы с изображения были выбраны библиотеки OpenCV[3] и Python Image Library (далее PIL). Для визуализации данных и их обработки использовались библиотеки matplotlib и numpy.

Ниже приведено описание работы, которая была проделана для реализации каждого из вышеуказанных этапов создания датасета.

### Создание анкеты

При выполнении этого этапа отрисовку анкеты было решено реализовать программными методами, используя встроенные методы библиотеки PIL и её форка pillow. Подобный подход позволяет сохранить координаты всех клеток анкеты, что сильно облегчает работу на этапе извлечения букв. Разработанный шаблон (рис. 1, ниже) имеет следующие элементы:

* буквы-образцы и клетки для написания рукописных букв
* черные квадраты по краям, необходимые для центрирования отсканированного изображения на этапе извлечения букв
* ARUCO – маркеры, находящееся на нижней части каждого листа. Они являются в некоторой степени аналогами QR-кодов или штрих-кодов и служат для кодирования численной информации. На каждом листе располагаются два маркера, один из которых содержит информацию о номере страницы, другой – уникальный номер респондента.

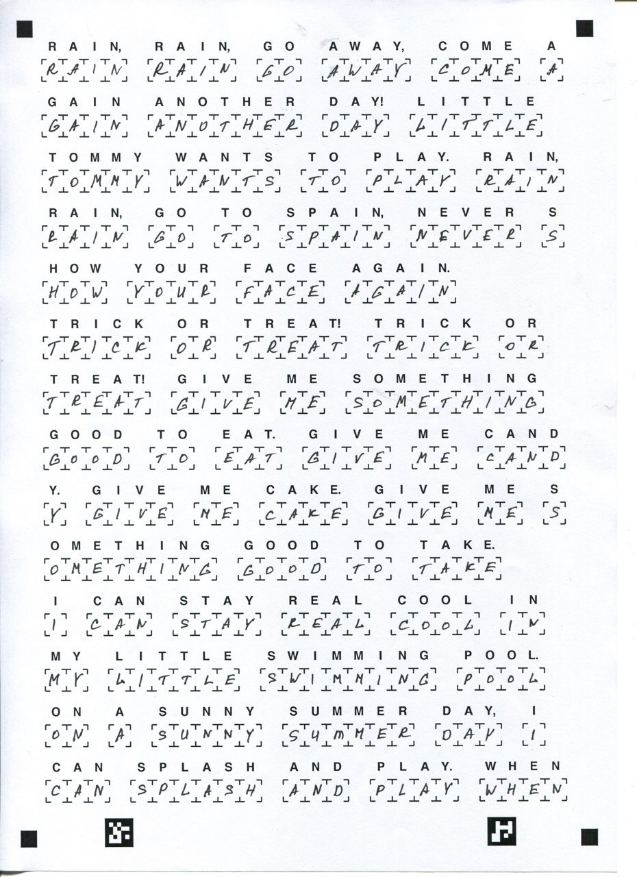
Последний может использоваться, например, если обнаружится, что респондент заполнил одну из страниц неправильно – в этом случае будет возможно удалить из выборки остальные заполненные им страницы анкеты.

Для отрисовки анкеты был реализован класс, обладающий методами для:

* инициализации страницы с автоматическим нанесением меток для центрирования и ARUCO – маркеров
* добавления ARUCO – маркеров, кодирующих дополнительную информацию
* отрисовки букв и клеток при передаче в метод символьной строки. Размер строки не ограничен, перевод курсора на следующую строку и страницу производится автоматически, новая страница инициализируется сразу со всеми необходимыми метками и маркерами. Во время отрисовки координаты всех клеток сохраняются в словаре вида {координаты клетки : буква, которая будет располагаться по этим координатам}, для каждой страницы - отдельно
* ручного перевода курсора на следующую страницу или же строку
* сохранения словарей координат для всех страниц в pickle – файл
* создания pdf – файла, содержащего все страницы

При инициализации экземпляра класса возможно задание произвольного размера страницы, размера клеток, размера шрифта, типа шрифта, цвета элементов.

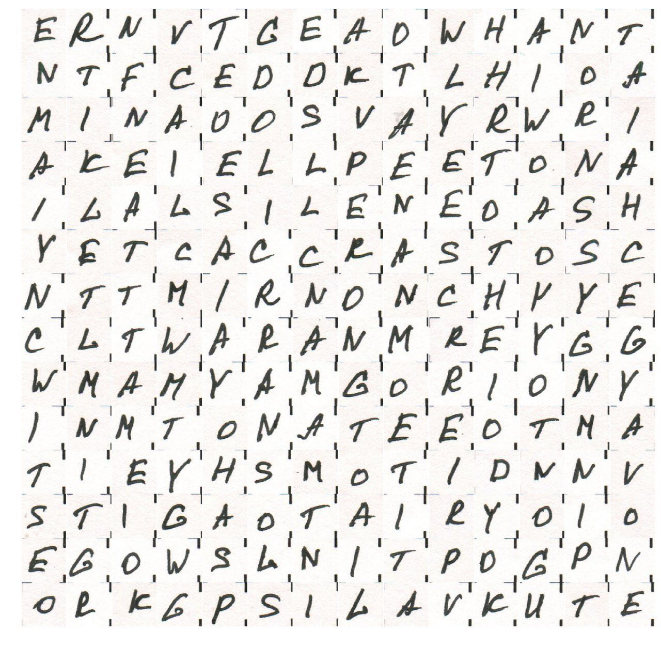
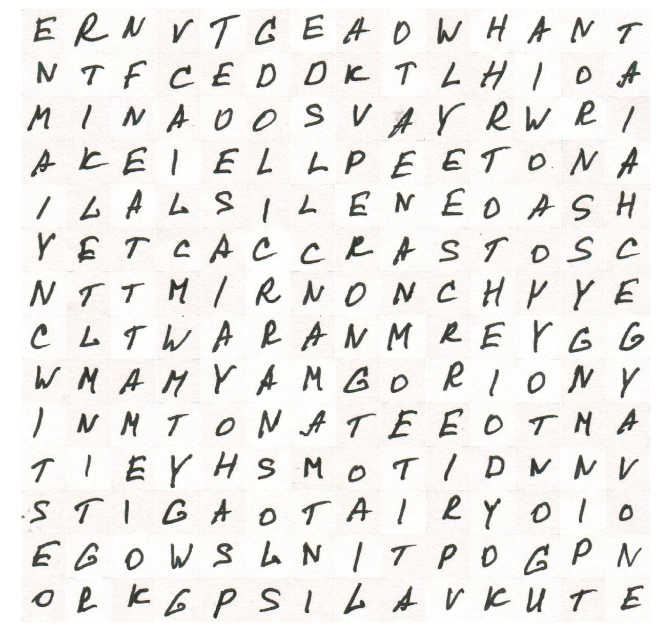
### Сканирование анкет и их обработка

После подготовки анкет распечатанные экземпляры были переданы 20 респондентам для заполнения (рис. 2). Сканирование проводилось на домашнем сканере, что, однако не сказалось на качестве полученного в конце работы тестового датасета. Также оцифровку можно проводить с помощью мобильных телефонов. Отсутствие высоких требований к оборудованию позволяет упростить работу с разработанной библиотекой и снижает порог вхождения для пользователей.

### Извлечение букв

На этапе формирования анкеты для каждой страницы был сформирован и сохранен словарь координат букв. Координаты определены в системе отсчёта анкеты, однако отсканированный документ имеет свою, не связанную с оригинальной систему отсчёта. Для того, чтобы извлечь буквы из клеток по известным координатам требуется, применив композицию сдвигов, поворотов и растяжений, привести одну систему координат к другой. Для решения этой задачи было выбрано применить реализованную в пакете OpenCV функцию getAffineTransform, которая, принимая три пары координат точек формата «начальные координаты точки – координаты, куда её необходимо переместить», возвращает матрицу аффинного преобразования, позволяющую провести необходимое приведение систем отсчёта. В качестве точек для преобразования были взяты центры центрирующих квадратов, найденных по следующему алгоритму – на первой стадии к изображению применили фильтр Canny, после чего было произведено заполнение контуров с помощью функции morphologyEx с ядром getStructureElement. К преобразованному таким образом изображению применили функцию findContours, и, аппроксимировав найденные контуры многоугольниками с помощью approxPolyDP, выделили все квадраты с заданным периметром. Подробно алгоритм описан в [4].

Несмотря на центрирование отсканированного документа и приведение систем отсчёта, погрешность, накопившаяся во время предыдущих операций, не позволила провести извлечение букв наивным методом. Была разработана специальная адаптивная функция извлечения, позволившая значительно улучшить качество полученных образцов (рис. 3а, 3б).

Работа адаптивной функции основывается на следующем принципе – она получает на вход координаты клетки, а затем, проверяя наличие артефактов разметки по краям извлечённого наивной функцией изображения, применяет композицию сдвигов по различным осям, нивелируя погрешность, и возвращает более «чистое» изображение. Алгоритм работы адаптивной функции является весьма стойким к различному шуму и искажениям.

### Формирование датасета

Экземпляры всех отсканированных документов были обработаны с применением вышеописанного алгоритма. Полученный датасет был сохранен в следующем формате – для каждого респондента была создана отдельная папка, в которой были созданы папки для всех типов букв. В эти папки были помещены извлечённые изображения букв.

Альтернативный формат, сохранение в котором было реализовано, подразумевает создание csv – файла со следующей структурой (по столбцам) – «тип буквы – идентификатор респондента – список интенсивностей всех пикселей изображения, каждый в своём столбце». Сформированные материалы были переданы для дальнейшей работы специалистам.

### Обучение алгоритмов ML

Полученный датасет был также использован для обучения распознающей рукопечатный текст модели. Для тестов были выбраны классификаторы, основанные на следующих методах:

* метод опорных векторов[5]
* метод k ближних соседей[6]
* лес решающих деревьев[7]

Также была проведена предподготовка данных, которая заключалось в выделении 50 главных компонент, а также последующем масштабировании данных в интервал [-1, 1]. Для обучения использовались реализации алгоритмов из пакета sklearn[8].

Разделение на тестовую и тренировочную выборку производилось двумя методами. Первый, наивный метод подразумевал случайное распределение объектов, другой, сбалансированный, исключал появление букв, написанных одним человеком в двух разных выборках. Точность моделей после обучения представлена в таблице 1.

|  |  |  |
| --- | --- | --- |
| Классификатор | Точность (наивный метод) | Точность (сбалансированный метод ) |
| SVM | 83.76% | 74.15% |
| kNN | 67.77% | 49.33% |
| Random Forest | 70.05% | 61.08% |



## Заключение

Вся описанная в отчёте работа была выполнена автором. В процессе разработки библиотеки автором были получены такие навыки, как:

* Python-разработка с использованием парадигм объектно-ориентированного программирования
* работа с графическими данными средствами библиотек PIL и OpenCV
* работа с алгоритмами компьютерного зрения с помощью библиотеки OpenCV

Автором также были приобретены знания о задачах и методах классического машинного обучения, существующих датасетах, их структуре, а также о всех этапах создания новых датасетов, содержащих графические данные. Автор научился вести разработку в понятной форме, пользоваться общими хранилищами для передачи информации. Практика являлась крайне полезной с точки зрения полученных предметных знаний и особенно с точки зрения полученного опыта реальной разработки.

Список литературы

1. https://ru.wikipedia.org/wiki/ Машинное\_обучение
2. http://www.image-net.org/challenges/LSVRC/
3. <https://docs.opencv.org/>
4. <https://pythontips.com/2015/03/11/a-guide-to-finding-books-in-images-using-python-and-opencv/>
5. *Nello Cristianini, John Shawe-Taylor.* An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. — Cambridge University Press, 2000.
6. *Christopher M. Bishop.* Pattern Recognition and Machine Learning
7. *Hastie, T., Tibshirani R., Friedman J.* Chapter 15. Random Forests // [The Elements of Statistical Learning: Data Mining, Inference, and Prediction](http://www-stat.stanford.edu/~tibs/ElemStatLearn/). — 2nd ed. — Springer-Verlag, 2009.
8. http://scikit-learn.org/stable/index.html