**СОДЕРЖАНИЕ**

1. ВВЕДЕНИЕ (5 стр.)

* В чем заключается задача? Почему это важно?
* В чем заключается наш вклад и новизна научно-прикладной работы?
* Может ли наше решение быть применено на практике?

2. ОБЗОР ЛИТ. ПО РАСПОЗНАВАНИЮ РУКОПИСНОГО ТЕКСТА (5 стр.)

3. МЕТОДОЛОГИЯ (15 стр.)

3.1. Сбор данных (5 стр.)

3.2. Выбор моделей детекции и распознавания (5 стр.)

3.3. Калибрация и валидация моделей распознавания (5 стр.)

3.4. Корректировка результатов распознавания на базе априорных знаний

4. ПРОЕКТИРОВАНИЕ СИСТЕМЫ И РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (20 стр.)

4.1. Графический пользовательский интерфейс

4.2. Выбор стека технологий

4.3. Архитектура системы

4.4. Разработка программного обеспечения

4.5. Валидация программного обеспечения

5.6. Характеристика программы и лицензия

6. ЗАКЛЮЧЕНИЕ (5 стр.)

**1. ВВЕДЕНИЕ**

*В чем заключается наша задача?*

Задача заключается в распознавании рукописного текста на почтовом конверте. Текст предполагается на любом из трех языков (казахский, русский, английский).

*В чем актуальность данной задачи?*

Вопрос распознавания рукописного текста является широко и активно изучаемым вопросом в научном мире. На данный момент не существует готовых решений, которые могли бы гарантировать высокую точность распознавания рукописного текста, что не позволяет большим компаниям, таким как Siemens (Рис. 1) или Blueshift (Рис. 2), внедрить ее в массовое производство. Их решения автоматической сортировки писем в основном построенны на распознавании печатного адреса, но не рукописного.



### Рисунок 1. Один из аппаратно-программных комплексов Siemens для сортировки писем.

### <https://www.siemens-logistics.com/en/mail-sorting/letter-sorting-and-sequencing>

Ранее, в АО «Казпочта» была внедрена система автоматизированной обработки почтовой корреспонденции от компании Siemens, которая предполагала решить вопрос с оптимизацией приема и сортировки почты, включая решение вопроса распознавания рукописного текста. В ходе опытной эксплуатации, было выявлено, что данная система оказалась несоответствующей текущим бизнес-процессам АО «Казпочта», и, на данный момент, является неработоспособной.

Причина неработоспособности данной системы может заключаться в том, что большинство писем, которые не имееют печатного адреса должны быть сортированы вручную (Рис. 3). Как показывает практика, ручная сортировка имеет множество недостатков. К ним относятся большие временные затраты, а также ошибки, связанные с человеческим фактором.

В данной работе мы хотели бы изучить потенциальные решения того, как можно было бы повысить эффективность ручной сортировки или же сократить усилия, которые она требует, что позволило бы сократить время обслуживания и повысить точность. Если говорить более конкретно, то мы рассматриваем задачу распознавания рукописного адреса как некий предварительный шаг, необходимый для дальнейшей автоматической сортировки.

Отталкиваясь от того, что распознавание рукописого адреса может быть использовано для банального автоматического или полу-автоматического ввода данных в систему почтовой службы. Имея адреса отправителя и получателя в электронном виде, оператор почтовых услуг мог бы представить их в любой удобной для него форме (штрихкод, печатный текст и т.д.). Допустим, оператор решает печатать электронный текст на стикерах, которые затем помещаются на письма, поверх рукописного текста. Этот предварительный шаг позволяет подготовить письма для дальнейшей автоматической сортировки. Следовательно, наши наработки могли бы быть потенциально использованы в комплекте с аппаратно-программным комплексом по сортировке писем в качестве модуля предобработки. В случае отсутствия автоматической сортировки, наш сервис мог бы просто ускорить процесс ручной сортировки письменной корреспонденции, так как бы исключил проблему набора текста вручную.

### 

### Рисунок 2. Vantage Sorting System, установленная в Минске.

### <http://zviazda.by/ru/news/20191009/1570601704-kak-sortiruyut-pochtovye-otpravleniya>

### Кі­ра Ера­шэ­віч — сар­та­валь­нік паш­то­вых ад­праў­лен­няў і дру­ка­ва­ных срод­каў ма­са­вай ін­фар­ма­цыі цэ­ха пісь­мо­вай ка­рэс­пан­дэн­цыі і гіб­рыд­най пош­ты вы­твор­час­ці «Мін­ская пош­та» РУП «Бел­пош­та».

### Рисунок 3. Ручная сортировка писем нестандартного размера или же с рукописным почтовым адресом

### <http://zviazda.by/ru/news/20191009/1570601704-kak-sortiruyut-pochtovye-otpravleniya>

Грубо письма можно классифицировать на:

* Входящую корреспонденцию
* Исходящую корреспонденцию

В первом случае, опреатору может понадобиться внести данные о письме в реестр (что оно принято в отделении) и доставить конечному адресату. Внесение данных в реестр, к примеру, может понадобиться когда письмо заказное или же необходимо указывать его передвижение и статус в онлайн системе трэкинга корреспонденции. Если у письма имеется штрихкод, то проблем не возникает и необходимо просто отсканировать штрихкод. Если же у письма нет штрихкода, то оператор вынужден вбивать данные в систему вручную, что может занять у него некоторое время и потенциально могут быть допущены ошибки в силу человеческого фактора. Здесь письма должны в основном быть распределены внутри страны. В зарубежной литературе подобную доставку называют доставка последней мили (“last-mile delivery”).

Во втором случае почтовому оператору необходимо вбить адрес в систему (в случае если метаданные письма вносятся в реестр компании и генерируется штрихкод или же трэкинговый номер). Обычно процесс вбивания адреса в ручную занимает относительно продолжительное время, а также допускаются множетвенные ошибки при наборе текста, т.к. не все операторы владеют навыками скоропечатания. Потенциально письма могут иметь пункт назначения за пределами страны. В этом случае письма пишутся или на русском (в случае пункта назначения в одной из стран СНГ), или на английском (в случае дальнего зарубежья). При локальной корреспонденции (внутри страны) адреса в преобладающем большинстве указаны или на казахском, или на русском.

Наше решение, построенное на распознавании рукописного адреса, могло бы быть полезно в обоих случаях. С использованием нашего сервиса провайдер почтовых услуг мог бы значительно ускорить процесс обслуживания клиентов, что автоматически привело бы к определенному увеличению качества обслуживания клиентов. В результате рейтинг компании и доверие клиентов к компании могли бы со временем возрасти.

*Как предположительно выглядит процесс сортировки?*

*[перефразировать][опциально]*

*Вся бумажная корреспонденция автоматически делится на стандартные (по габаритам) и нестандартные письма. После этого их автоматически укладывают лицом в одну сторону с учетом направления текста в адресной части, затем машина наносит на письма календарный штемпель.*

*Рассортированные по размерам письма складывают в контейнеры и отправляют в машину для кодирования. В начале процесса кодировочные аппараты преобразуют цифровой или буквенный адрес в условный код, который печатается на письме. Далее закодированную информацию о почтовом адресе сканируют устройством для считывания символов (оборудование в том числе поддерживает распознавание кириллицы). В случае невозможности распознать индекс или адрес отсканированное изображение поступает на монитор операторам, которые вводят информацию вручную. После этого письма автоматически разделяют по адресным ячейкам. Выемка ящиков с отсортированной почтой и подготовка документов для отправки осуществляется вручную.*

*На участок ручной сортировки писем поступает почта, которая не подлежит обработке в автоматической машине, например, из-за дефекта упаковки, неправильно написанного адреса или недопустимого вложения внутри письма (не с точки зрения цензуры, а потому что могут повредить машину или упаковку). Корреспонденция сортируется по индексам. В зависимости от адреса оператор раскладывает их по определенным ячейкам.*

*На участке сортировки обрабатывают около 8 тыс. посылок за час. Оператор устанавливает их на линию, после чего проходит взвешивание. Если вес допустим по нормативам (до 20 кг), посылка считывается сканером и попадает на ленту, если нет — машина останавливается, и подключается ручной труд.*

*После того как посылка прошла по конвейерной ленте, она выпадает в определенную ячейку, которой присваивается штрих-код. Это сделано для того, чтобы посылки не терялись в пути следования. Вся информация об отправлении, таким образом, попадает в единую систему Почты России.*

*Рассортированную почту загружают в машины и развозят по отделениям. Грузовики приезжают сюда 2 раза в сутки: утром и вечером. Посылки и письма заносят в разные комнаты и регистрируют. В отделе доставки письма вручную штампуют, нанося дату, время приема и координаты почтового отделения. Далее оператор раскладывает их по ящикам с номером участка, откуда газеты и письма забирают почтальоны.*

*Заказные письма и посылки взвешивают и с помощью прибора со штрих-кодом заносят в специальную программу. Она позволяет отслеживать статус посылки (посмотреть его можно на официальном сайте Почты России, при условии, что вы знаете ваш номер почтового отправления).*

*Срок получения письма может сократить сам получатель, если будет придерживаться определенных правил. Корректно указать индекс: 30% отправлений задерживаются из-за орфографических ошибок и опечаток.*

*В чем заключается вклад и новизна нашей научно-прикладной работы?*

Во-первых, наш научный вклад заключается в том, что был собран и аннотирован довольно обширный датасет рукописных слов на казахском и русском языках. По количеству собранных и аннотированных слов, насколько нам известно, нашему датасету нет аналогов.

Во-вторых, для сбора данных были использованы новейшие технологии, такие как использование телеграм ботов для аннотации данных. Другими словами, подобный метод аннотации рукописного текста практически не применялся до этого. Он показал свою эффективность на практике. Он позволяет ускорить скорость аннотации, посредством расширения аудитории людей готовых посвятить несколько минут на аннотацию слов.

В-третьих, современые модели компьютерного зрения по разпознаванию были адаптированы и калиброваны на собранном датасете. В основном эти модели применялись в общем контексте распознавания объектов в различных сценах, печатного или же рукописного английского текста. Однако, ни одна из современных моделей не применялась конкретно для распознавания рукописного почтового адреса. Тот факт, что эти модели были целенаправлено калиброваны на распознавание казахского и русского алфавита увеличивает значимость вклада и подчеркивает его важность.

В-четвертых, в данном проекте были использованы априорные знания, а имеено контекст задачи (структура лицевой части конвертов и содержание текста). Мы твердо уверены, что использование априорных знаний о положении текста, свойственном в почтовой кореспонденции, содержании каждого из полей дает преимущество перед более общей задачей распознавания рукописного текста. Другими словами, используя эти знания, точность распознавания может быть значительно повышена. В данной работе предложен эффективный метод коррекции результатов распознавания, который позволяет сократить пространство поиска, опираясь на априорные знания.

*Какой ожидаемый результат?*

В результате работы будет разработан прототип (proof of concept) сервиса распознавания рукописного адреса. Он будет представлен в виде веб-сервиса. В его функционал будут входить следующие аспекты:

1. Захват изображения
2. Предобработка изображения (выделение области конверта и компенсация пространственной трансформации)
3. Автоматическое определение положения полей отправителя и получателя
4. Возможность коррекции автоматического определения положения полей отправителя и получателя
5. Сегментация адреса по словам
6. Возможность коррекции сегментации на слова
7. Распознавание рукописных слов
8. Постобработка распознанных слов
9. Возможность ручной коррекции распознанных слов

При необходимости сервис может быть адаптирован и интегрирован в бизнес процессы КазПочты.

**2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ**

Распознавание рукописного адреса является комплексной задачей. Перед тем как начать описание архитектуры системы и графический пользовательский интерфейс, позвольте нам кратко описать этапы, из которых весь процесс распознавания состоит и явно объяснить причины, по которым те или иные решения были приняты.

В контексте нашей формулировки весь процесс можно представить следующим образом:

1. Выполнение снимка (сканирование или фотографирование)
2. Распознование адреса
   1. предобработка снимка (выравнивание)
   2. локализация областей снимка с даными отправителя/получателя
   3. сегментация целевых областей на слова
   4. распознование сегментированных слов
   5. коррекция результатов распознования слов

Графически процесс распознавания рукописного адреса представлен на Рис. 4.

Рисунок 4. Общие этапы процесса распознавания рукописного текста.

Как показано на Рис. 4, сначала необходимо сделать снимок лицевой стороны конверта. Здесь возникает множество вопросов относительно того как именно делать снимок – с помощью камеры или же сканера. Сканер не подходит для работы в реальном времени, поэтому единственным вариантом остается камера. Далее, как камера и объект должны быть расположены друг относительно друга, какую камеру выбрать (какими характеристиками она должна обладать), какое разрешение картинки должно быть получено, на каком расстоянии от камеры должен находиться объект, с какой частотой должны делаться снимки, держать камеру в руках или же закрепить и т.д. Если камера статична и объект динамичен, то это может привести к тому, что в результате у нас будут снимки с различными перспективными искажениями (напомним, что желательно, чтобы плоскость конверта была перпендикулярна фокальной оси камеры), изображение может содержать блур, т.к. руки оператора могут совершать микродвижения при выполнении снимка или же конверт поднесен черезчур близко к камере. Практически эквивалентная ситуация возникает при станичном конверте (к примеру, он лежит на столе) и динамичной камере. Камера может двигаться или может находиться черезчур близко к объекту, что потенциально может вызвать размытие результирующего снимка. Дополнительно возникает вопрос заднего фона. Что если на заднем фоне конверта будут располагаться другие документы с рукописным текстом? Как мы определим целевую область на картинке в этом случае? Другим немаловажным аспектом является освещение. Как должен быть выставлен объект, чтобы он был достаточно освещен и не создавались различные артефакты типа теней? Мы видим, что с точки зрения снимков идеально было бы создать такие же условия как в сканере – нет заднего плана, освещение всегда одинаковое, положение камеры относительно объекта фиксировано, фокусное расстояние также фиксировано. С учетом выше сказанного, для того чтобы создать такие же условия имеет смысл фиксировать положение камеры, освещения и объекта относительно друг друга. Это позволило бы создать однородные условия при выполнении снимка. Дополнительно, конверты могут быть разных размеров, поэтому в более общем случае нам могут быть неизвестны габариты. Конечно, в интерфейсе сервиса можно указать индикатор для области, куда конверт должен быть помещен. Однако этот подход может значительно сократить практичность нашего решения.

После того как снимок выполнен начинается основной функционал нашего сервиса. Как показано на Рис. 5а, он включает несколько этапов.

Во-первых, входное изображение может требовать предварительной обработки перед тем, как будет использовано. Может быть необходимо определить область, которая соответствует конверту (Рис. 5б). В случае с однородным задним фоном, этот этап не так критичен. В случае с неоднородным задним фоном, этот этап критичен. К примеру, потенциально снимок может быть сделан на фоне доски с надписями или же на столе на фоне других документов с текстом. Для более точного распознавания рекоммендуется обрезать области, не принадлежащие конверту, чтобы исключить любой паразитный текст на раннем этапе. После того как область конверта определена, может оказаться, что конверт имеет некоторое вращение (Рис. 5б). Необходимо компенсировать это вращение. Потенциально это может увеличить точность сегментации и, как следствие, точность распознавания. За это ответсвеннен модуль предобработки снимка. Также может быть необходимо применить различные сглалживающие фильтры для сглаживания шумов.

|  |  |
| --- | --- |
|  |  |
| (а) | (б) |

Рисунок 5. Основной функционал сервиса распознавания рукописного адреса.

Во-вторых, после того как изображение было подготовлено для использования модулем предобработки, снимок далее направляется в модуль определения областей конверта с целевым текстом. К этим областям относятся те участки снимка конверта, где находятся данные отправителя и получателя. Здесь необходимо будет использовать эвристики по поводу того, как эти области располагаются относительно друг другу. К примеру, в постсоветских странах принято писать данные отправителя в верхнем левом углу конверта, а данные получателя – в правом нижнем углу. Однако, эти допущения могут не работать для других стран. Другим способом может быть нахождение ключевых слов или фраз. Однако, они также могут отсутствовать и в общем случае подобное решение может не работать. Дополнительно, конверты очень сильно отличаются друг от друга: по размеру, цвету и содержанию. К примеру, как показано на Рис. 6, на конвертах может быть разное количество марок, паразитный текст или положение областей получателя-отправителя могут быть смещены или вообще отсутствовать.



Рисунок 6. Конверт с большим количеством паразитного текста и печатей.

Следовательно, невозможно обеспечить 100% точность работы системы. В связи с этим, в случае некорректного автоматического срабатывания системы, мы предоставляем оператору возможность вручную указывать область снимка, где находится область получателя. В этих условиях система автоматического определения положения целевого текста может не сработать. Это недопустимо для прикладных решений. Другими словами, у пользователя должна быть возможность вносить коррективы в результаты автоматической локализации. Как показано на Рис. 7, мы закладываем этот функционал в наш сервис.

Рисунок 7. Определение областей с целевым текстом.

Следовательно, пользователь сможет, в зависимости от ситуации, переключаться с автоматического режима определения целевых областей с текстом на ручной или же полуавтоматический.

На Рис. 8 показан модуль распознавания текста. Он базируется на калиброванных моделях, описанных ниже. Как было упомянуто в предыдущих главах, адреса на конвертах, которые оказываются в отделении почты Казахстана, могут быть на одном из трех языков: казахский, русский и английский. Для каждой из этих моделей в нашем модуле имеется натренированная модель. Дополнительно, имеется модель которая калибрирована на всех трех языках. Однако ее точность, аналогично точности общей модели, предоставляемой платформой Tesseract, намного ниже, чем отдельных моделей. Причина заключается в том, что обучение модели на выборки разных языков может внести некоторую неопределенность в модель, так как на одни и теже входные данные у модели будет несколько эквивалентных ответов. Отсюда следует, что желательно, чтобы язык, на котором адреса написаны, был определен перед началом непосредственного распознавания слов. Определение языка может быть осуществленно автоматически или же вручную. Оператор может переключиться на ручной режим определения языка в случае неверного результата автоматического определения языка.

Рисунок 8. Модуль распознавания сегментированных слов.

Итак, письменная корреспонденция может быть как внутренней, так и международной, поэтому нам необходимо рассматривать несколько языков: казахский, русский, английский. Для большей точности рекоммендуется иметь отдельную модель распознавания для каждого языка.

Допустим мы разбили область снимка с данными получателя на слова. Предположим, что в этом наборе слов обязательно есть название страны. В случае, если страна не указана, то по умолчанию корреспонденция считается внутренней.

Внутренняя переписка может вестись на казахском и русских языках. Переписка по странам СНГ может вестись на русском языке. Международная переписка – на английском.

С учетом этого допущения, для определения языка нам необходимо иметь словарь ключевых слов для каждого языка. В данном случае, ключевыми словами будут считаться названия стран ( слов если учитывать сокращения и т.д.).

Следовательно, мы пробегаемся по всем картинкам, распознавая слова с использованием одной из моделей распознавания, и сравниваем с каждым словом в базе данных ключевых слов. С использованием определенной метрики выбираем наиболее вероятную модель распознавания языка.

Рисунок 9. Схематичное представление принципа выбора языка.

Допуская, что модель распознавания корректно выбрана и страна корректно определена. Для каждой страны мы храним словарь ключевых слов, который включает в себя названия провинций, областей, городов (<1000 слов). Этого должно быть достаточно для сортировки писем для межгородских сообщений.

Следовательно, для коррекции результатов распознавания мы можем находить степень соответсвия ключевым словам и заменять соответствующими словами.

В силу того, что мы работаем над прототипом, а не над готовым программным продуктом, некоторый описанный выше функционал не будет имплементирован в рамках проекта. К примеру, в рамках проекта мы сконцентрируемся только на одном языке - казахском. С нашей точки зрения, для наглядной демонстрации возможности нашего сервиса этого должно быть достаточно.

**ГРАФИЧЕСКИЙ ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС**

Главная страница приложения предоставляет только возможность загрузки файла конверта, данное решение было принято на основе рассуждения о том, что на данном этапе основной целью было разработать прототип приложения без учета функционала когда будут подаваться множество изображений потоком (Рис. 10).

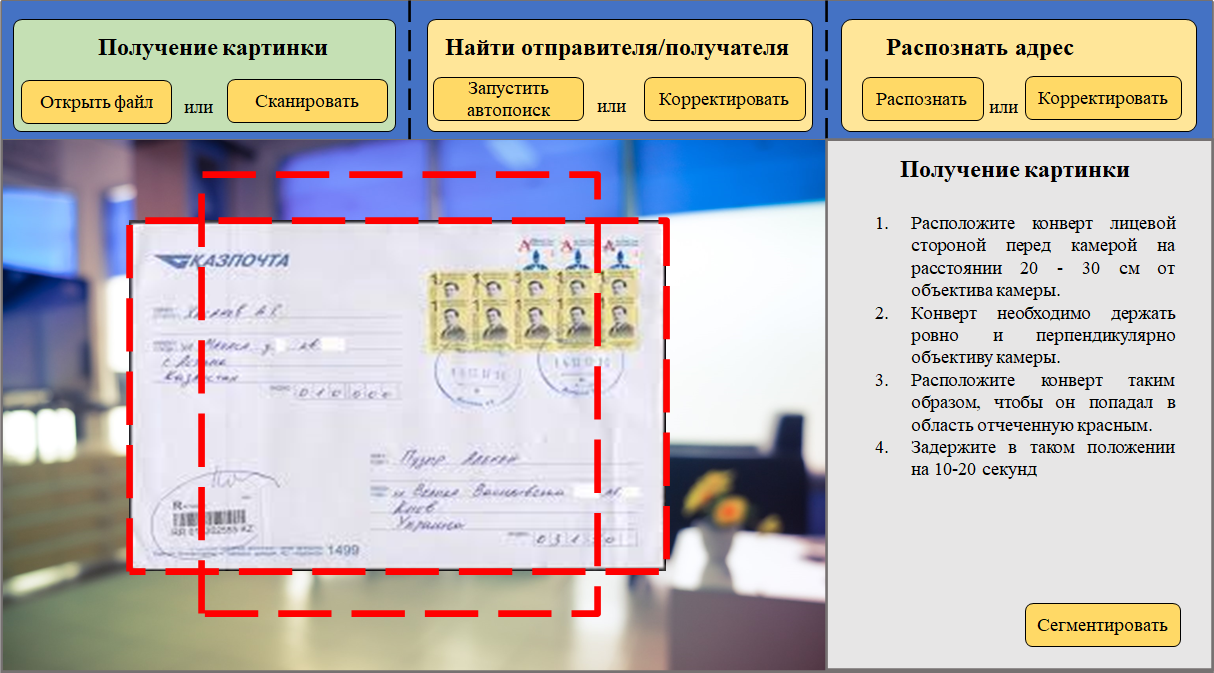


Рисунок 10. Создание снимка. При выполнении снимка оператору требуется расположить конверт на таком расстоянии от камеры, чтобы область конверта полностью помещалась в указанной пунктирными линиями области. Для инициации снимка необходимо зафиксировать конверт в таком положении на 10-20 секунд, после чего сервис автоматически произведет снимок.

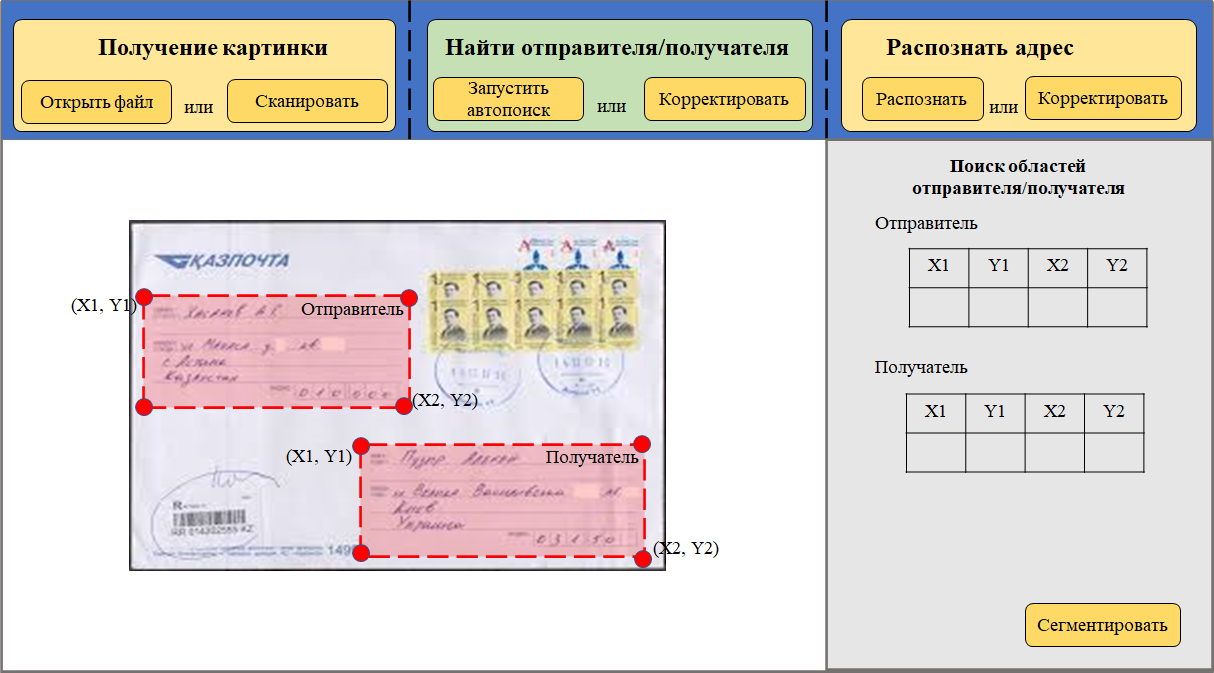


Рисунок 11. Сегментация областей конверта, где находятся данные получателя и отправителя. Каждая область определяется прямоугольником, который задется двумя вершинами. Для коррекции результатов оператору достаточно перетащить в нужную позицию круглые манипуляторы.

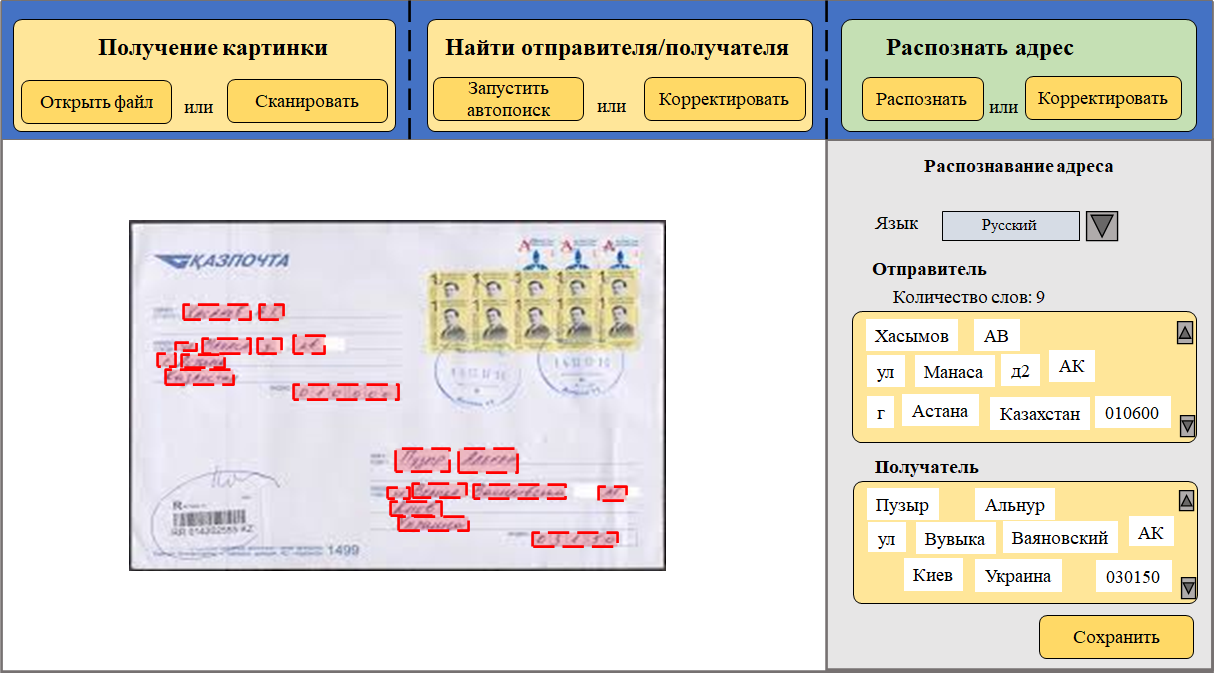


Рисунок 12. Вывод количества найденных слов и список распознанных слов. Для коррекции автоматического распознавания можно производить в самих ячейках слов.

**СТЭК ТЕХНОЛОГИЙ**

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

Python — активно развивающийся язык программирования, новые версии с добавлением/изменением языковых свойств выходят примерно раз в два с половиной года. Python поддерживает динамическую типизацию, то есть тип переменной определяется только во время исполнения. Поэтому вместо «присваивания значения переменной» лучше говорить о «связывании значения с некоторым именем».

Согласно данным опроса разработчиков Python в 2019, Django и Flask являются самыми популярными веб фреймворками среди разработчиков. Оба имеют собственные преимущества и недостатки. В то же время оба хорошо задокументированы.

Flask и Django – оба взрослые, расширяемые веб фреймворка, которые, в корне своем, предоставляют аналогичный функционал в обработке запросов, поддержке документов, но различаются в масштабе ответственности.

Большая часть отличий между двумя фреймворками проистекают из разных подходов, остальные – из отличных основных проектных решений. Вот небольшой список ключевых различий, которые могут повлиять на ваше решение:

* Объект Request – Flask использует локальные потоки, а Django передает запрос там, где это нужно.
* Формы – Django доступен со встроенными формами, которые интегрируются с ORM и админкой сайта. Flask не поддерживает формы по умолчанию, но вы можете использовать WTForms, чтобы заполнить этот пробел.
* Базы данных – Django доступен со встроенной ORM и системой миграции, которая может управлять базами данных. Flask не может этим похвастаться, однако есть инструменты, такие как SQLAlchemy, которые предоставляют аналогичный функционал (или даже больше).
* Аутентификация и привилегии пользователям – Django предоставляет приложение аутентификации, которое предоставляет реализацию по умолчанию для пользовательского управления и привилегий. Flask предоставляет безопасные куки в качестве инструмента вашей собственной реализации.
* Панель администратора – Django включает в себя полностью интегрированный админ-интерфейс для управления данными приложения. Flask не имеет таких функций, но Flask-Admin – очень популярное расширение, которое можно использовать для создания аналогичного административного инструмента.

На сайте Flask его описывают как “микрофреймворк для Python, основанный на Werkzeug, Jinja 2 и отличных решениях” и “веб-разработка, по капле за раз”. Еще раз, это дает вам хорошее представление о пространстве, которым Flask пытается заполнить переполненный мир веб-фреймворков Python.

В частности, Flask может вести себя как мини-фреймворк на Python, который хорошо обрабатывает ряд вещей, при этом оставляет место для решений, которые вы хотите принять при постройке собственного веб-приложения.

Django и Flask содержат сервера разработки, которые делают процесс создания веб приложения более удобным и быстрым. Они содержат функции, которые принято ожидать от взрослого веб-фреймворка, такие как возможность обработки запросов, подача статических файлов (для разработки), и обнаружить изменения в вашем коде для автоматического перезапуска, чтобы ваши изменения вступили в силу.

В данном проекте выбор пал в пользу Flask, так как он является более легковесным и быстрым в развертывании, в то же время в широком функционале Django мы не нуждаемся.

В качестве основной цветовой схемы были выбраны цвета Казпочты: синий, белый, желтый.

Следует отметить,что для верстки фронтенда использовался фреймворк Bootstrap, который ускоряет верстку сайтов, предоставляя широкий спектр инструментов стилизации кнопок, навигационной панели, сетки.

Bootstrap – это CSS фреймворк, который изначально создавался для внутреннего использования компанией «Twitter» с рабочим названием «Twitter Blueprint», но в итоге был опубликован в открытый доступ и стал хорошим набором инструментов для front-end разработки под названием «Bootstrap». Официальный сайт фреймворка находится по адресу getbootstrap.com.

Преимущества фреймворка Bootstrap:

* Высокая скорость разработки макетов страниц сайта. Bootstrap содержит огромный набор готовых решений и элементов.
* Кроссбраузерность и адаптивность сайта. Все элементы фреймворка адаптивны под все устройства и корректно отображаются во всех современных браузерах.
* Легкость в использовании. Даже человек, имеющий базовые знания о HTML и CSS, может свободно создавать web-страницы с использованием фреймворка.
* Простота в обучении. У Bootstrap очень хорошая документация с большим количеством примеров готового кода.

Недостатки Bootstrap:

* Кода обычно в библиотеке написано больше, чем если бы вы написали при разработке с нуля. Потому что когда вы делаете самостоятельно, вы реализуете только необходимый функционал и все. В Bootstrap же есть все на все случаи жизни. Даже то, что вам может не пригодиться. Но опять же, эта проблема очень легко решается тем, что вы можете сами выбирать, какие компоненты фреймворка загрузить в css-файл. Например, вы вообще можете скачать только сетку, а все остальное делать самостоятельно.
* Шаблонный дизайн. Да, действительно, я часто захожу на разные сайты и вижу там одинаковые кнопки. И я знаю, что они сделаны в Bootstrap, потому что уж слишком это очевидно. Но и эта проблема легко решается, потому что она будет существовать только в том случае, если вы будете использовать только готовые компоненты фреймворка и ничего никогда не кастомизировать под себя.

**СБОР ДАННЫХ ДЛЯ КАЛИБРАЦИИ МОДЕЛЕЙ РАСПОЗНАВАНИЯ**

**Сбор данных с использованием форм**

Нами были созданы специальные формы для сбора данных. Несколько примеров подобных форм представлены на Рис. 11. В основном эти формы использовались для сбора образцов рукописного написания слов на русском языке. Условно все формы можно разделить на 3 типа в зависимости от содержания:

1. Для первого набора данных был определен набор данных содержащий в себе все названия крупных и мелких городов Казахстана, областей и районов Казахстана, классификация населенных пунктов, сокращенные названия.

2. Для второго набора данных был определен минимальный набор слов, который включает названия городов и областей Казахстана, а также название республик СНГ.

3. Для третьего набора данных были отобраны поэмы, включающие различный набор слов русского языка.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| (а) | (б) | (в) |

Рисунок 11. Образцы заполненных форм: (а) форма с названиями городов; (б) форма с случайным текстом; (в) форма с алфавитом.

Статистика по количеству присутствия символов в датасете представлена на Рис. 12 и в Таблице 1.

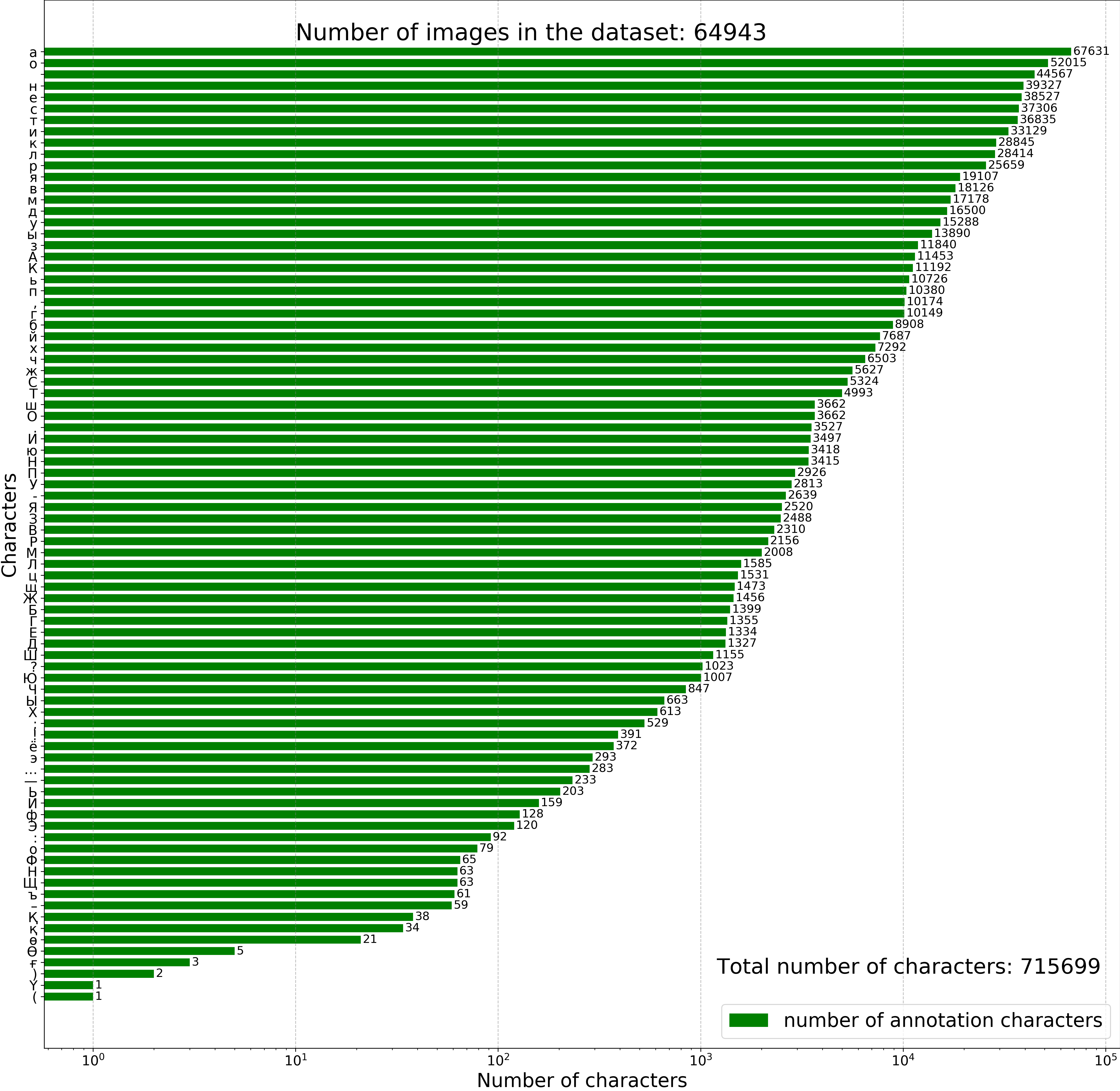


Рисунок 12. График распределения букв в базе данных.

Как видно из Рис. 12 и Таблицы 1, датасет покрывает большинство символов в достаточной степени. Оданко стоит отметить, что данная база данных касается в основном слов на русском языке. К примеру, это видно из того, что специфические казахские буквы практически не представлены в датасете (Таблица 1).

На данный момент, нам удалось собрать более 700 форм первого и второго типов (т.е., название городов Казахстана и стран мира) и около 900 форм третьего типа (т.е. с поэзией).

Таблица 1 – Табличное представление распределения букв в базе данных

| № | Буква | Количество  символов |
| --- | --- | --- |
|  | а | 67631 |
|  | ә | - |
|  | б | 8908 |
|  | в | 18126 |
|  | г | 10149 |
|  | ғ | 3 |
|  | д | 16500 |
|  | е | 38527 |
|  | ё | 372 |
|  | ж | 5627 |
|  | з | 11840 |
|  | и | 33129 |
|  | й | 7687 |
|  | к | 28845 |
|  | қ | 34 |
|  | л | 28414 |
|  | м | 17178 |
|  | н | 39327 |
|  | ң | - |
|  | о | 52094 |
|  | ө | 21 |
|  | п | 10380 |
|  | р | 25659 |
|  | с | 37306 |
|  | т | 36835 |
|  | у | 15288 |
|  | ұ | - |
|  | ү | - |
|  | ф | 128 |
|  | х | 7292 |
|  | һ | - |
|  | ц | 1531 |
|  | ч | 6503 |
|  | ш | 3662 |
|  | щ | 1473 |
|  | ъ | 61 |
|  | ы | 13890 |
|  | і | - |
|  | ь | 10726 |
|  | э | 293 |
|  | ю | 3418 |
|  | я | 19107 |
|  | А | 11453 |
|  | Ә | - |
|  | Б | 1399 |
|  | В | 2310 |
|  | Г | 1355 |
|  | Ғ | - |
|  | Д | 1327 |
|  | Е | 1334 |
|  | Ё | - |
|  | Ж | 1456 |
|  | З | 2488 |
|  | И | 3497 |
|  | Й | 159 |
|  | К | 11192 |
|  | Қ | 38 |
|  | Л | 1585 |
|  | М | 2008 |
|  | Н | 3478 |
|  | Ң | - |
|  | О | 3662 |
|  | Ө | 5 |
|  | П | 2926 |
|  | Р | 2156 |
|  | С | 5324 |
|  | Т | 4993 |
|  | У | 2813 |
|  | Ұ | - |
|  | Ү | 1 |
|  | Ф | 65 |
|  | Х | 613 |
|  | Һ | - |
|  | Ц | - |
|  | Ч | 847 |
|  | Ш | 1155 |
|  | Щ | 63 |
|  | Ъ | - |
|  | І | - |
|  | Ы | 663 |
|  | Ь | 203 |
|  | Э | 120 |
|  | Ю | 1007 |
|  | Я | 2520 |
|  | пробел | 44567 |
|  | - | 2639 |
|  | – | 59 |
|  | — | 233 |
|  | : | 92 |
|  | ; | 529 |
|  | ! | 391 |
|  | ? | 1023 |
|  | , | 10174 |
|  | . | 3527 |
|  | … | 283 |
|  | ( | 1 |
|  | ) | 2 |

Преимущество данного подхода заключалось в том, что не требовалось прикладывать дополнительных усилий для аннотации слов, т.к. формы были спроектированы таким образом, что позиции сегментов изображения однозначно пределяли их содержание.

Данный подход имел два больших недостатка. Во-первых, количество слов в формах было строго ограничено. Ограниченность слов в датасете сказалась на несбалансированном представлении различных букв. Некоторые буквы, к примеру, «ы», «ь», «ц», «э» были значительно меньше представлены в выборке, чем остальные буквы. Несбалансированность датасета не позволяла калибровать модель до более высокой степени точности распознавания. Во-вторых, заполнение форм требывало непосредственного активного участия респондентов. Другими словами, респонденты должны были заполнять формы. В силу времяемкости процесса заполнения, каждый респондент мог заполнить не более 3-5 форм.

Следовательно, мы задались следующим вопросом: можно ли использовать уже имеющиеся рукописные документы в качестве данных для калибрации моделей распознавания? Основной проблемой в данном методе является то, каким образом аннотировать неструктурированный случайный текст. При стандартном подходе это могло быть неэффективно. Было найдено довольно элегантное решение, которое описано ниже.

**Сбор данных с использованием случайного текста**

В основном этот метод был использован для сбора датасета на казахском языке. Было отсканировано около 2000 страниц А4 экзаменационных листов студентов на казахском и русском языках. Далее, полученные изображения были сегментированы по словам. Из 2000 страниц было получено 180000 слов. Планируется отсканировать еще около 4000 страниц. Для аннотации слов был запущен телеграм бот. Принцип его работы заключается в том, что каждому участнику, при наборе определенной команды, показывается картинка, содержащая рукописное слово. Задача пользователя – написать набрать слово и отправить в телеграм канал. Бот эту информацию ассоциирует со словом и сохраняет в базе данных.

Данный подход имеет множество преимуществ. Во-первых, респондентам больше не нужно тратить время на написание текста. Во-вторых, так как текст случайный, то и вероятность получить более сбалансированный текст возрастает. Обратите внимание, что данный подход подразумевает, что респонденты не флудят и размечают слова правильно.

**СБОР ДАННЫХ ДЛЯ ВАЛИДАЦИИ СИСТЕМЫ**

Как это отображено в названии проекта, проект носит научно-прикладной характер. Изначально он предназначался для нужд конкретной компании, а именно КазПочты. Предполагалось, что наше решение будет валидировано на базе данных полученных с КазПочты. В качестве валидационных данных нам нужны были снимки лицевой части конвертов. Договоренность с руководством компании была достигнута еще до начала проекта. Однако, по разным непреодолимым обстоятельствам взаимодействия не получилось. В связи с чем нам пришлось искать альтернативные варианты сбора данных для валидации системы. А имеено, были собраны снимки на казахском и русском языках, находящихся в свободном доступе. Дополнительно, мы отсканировали имеющиеся у нас конверты. В общей сложности у нас оказалось NNN количество конвертов. Для более маштабной валидации системы доступ к прототипу может быть предоставлен по требованию.

**ЗАКЛЮЧЕНИЕ**

*Что удалось реализовать на данный момент? Что еще предстоит сделать?*

Написано несколько статей.

Собран и аннотирован приличный датасет рукописного текста на русском и английском языках.

За это время отработана и отшлифована методология сбора данных. Данный процесс по мере возможности был автоматизирован. Для сбора данных были использованы новейшие технологии и сервисы. К примеру, были отсканированы контрольные работы студентов, с использованием ранее написаных модулей эти работы были сегментированы на слова. Далее был написан telegram bot, который высылает картинку слова, а пользователь печатает в ответ слово, которое там написано. Таким образом мы аннотируем данные.