# **ВВЕДЕНИЕ**

1. СУЩЕСТВУЮЩИЕ РЕШЕНИЯ РАСПОЗНОВАНИЯ ТЕКСТА
   1. СКРЫТЫЕ МАРКОВСКИЕ МОДЕЛИ (HMM)
   2. СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ (CNN)
   3. РЕКУРЕНТНЫЕ НЕЙРОННЫЕ СЕТИ (RNN)
2. РАСПОЗНАВАНИЕ РУКОПИСНОГО АДРЕСА
   1. ОСНОВНАЯ ИДЕЯ
   2. АРХИТЕКТУРА СИСТЕМЫ
3. СБОР ДАННЫХ
   1. **СУЩЕСТВУЮЩИЕ БАЗЫ ДАННЫХ РУКОПИСНОГО ТЕКСТА**
   2. СБОР И ХРАНЕНИЕ ДАННЫХ
   3. БАЗА ДАННЫХ КЛЮЧЕВЫХ СЛОВ
   4. ОБРАЗЦЫ РУКОПИСНОГО АЛФАВИТА
   5. СБОР СКАНОВ ЛИЦЕВЫХ СТОРОН КОНВЕРТОВ
   6. СПОСОБЫ СБОРА ДАННЫХ
4. ЗАКЛЮЧЕНИЕ
5. ЛИТЕРАТУРА

# ****1 ВВЕДЕНИЕ****

Перед нами поставлена задача разработки и внедрения системы по распознанию рукописных адресов письменной корреспонденции с использованием машинного обучения.

Если описывать весь процесс работы конечной системы от начала до конца, то он будет состоять из следующих шагов:

* **Письма помещаются на движущийся конвейер лицевой стороной вверх**
* **На определенном участке конвейера делается снимок с определенной периодичностью**
* **Система обрабатывает снимок и выдает адреса отправителя и получателя**
* **В зависимости от адреса получателя письмо сортируется.**

Систему, подобную описанной выше, предлагал университет Буфало в лице Саргура Шрихари и Эдварда Куберта в статье «Integration of Hand-Written Address Interpretation Technology» в 1997 году [1].

На данном этапе наших исследований мы опустим аппаратную часть аппаратно-программного комплекса, и сконцентрируемся на программной составляющей.

Задача распознавания рукописного адреса (offline handwritten address recognition) является частным случаем распознавания рукописного текста (Off-line Cursive Word Recognition, CWR). Здесь основные различия заключаются в том, что, во-первых, множество слов для распознавания ограничено словами, которые могут встретиться в адресах; во-вторых, рукописный текст пишется на монотонном фоне, что значительно облегчает процесс сегментации. Однако остальные аспекты распознавания текста являются общими и их можно рассматривать независимо от конкретной сферы применения. К примеру, изображение конверта, так же как и изображение любого другого рукописного документа, не может напрямую быть использовано для распознавания адреса, так как система сначала должна определить место, где находится текст; отделить его от заднего фона; сегментировать по словам текст; и нормализовать слова таким образом, чтобы они стали свободны от пространственной трансформации. Только после этих процедур данные можно использовать для построения дескрипторов, которые являются входными данными в модели распознавания.

Мы изучили существующие решения и определили наиболее эффективные на сегодняшний день подходы. В главе 2 мы опишем этапы работ, которые необходимы для распознавания текста, а именно процесс сбора данных, обработка данных, сегментация снимков, выбор подходящих дескрипторов. В главе 3 будет описаны наиболее часто используемые в данном контексте модели распознавания, а именно модели на основе скрытых марковских моделей (HMM) [], гибридных марковских моделей (Hybrid HMM) [], а также сверточных и рекуррентных нейронных сетях []. В главе 4 мы опишем выбранную нами модель и приведем промежуточные результаты. Глава 5 содержит заключение.

Весь процесс работы можно разбить на несколько этапов, над которыми можно работать параллельно:

1. Сбор данных
2. Определение области с текстом на снимке
3. Сегментация текста
4. Распознавание слов

В данном отчете мы ограничимся описанием процесса сбора данных и распознавания слов. Сопутствующие задачи по определению области с текстом и сегментация текста на строки и слова мы планируем предоставить со следующим отчетом.

**2. СУЩЕСТВУЮЩИЕ РЕШЕНИЯ РАСПОЗНОВАНИЯ ТЕКСТА**

**После того, как данные для обучения модели распознавания рукописных слов собраны необходимо выбрать концептуальный способ решения задачи. К примеру, задачу распознавания слов можно рассматривать как задачу посимвольного распознавания (OCR) [3,4,5,6,7,8,9] или же как задачу распознавания по словам (OWR). Если рассматривать задачу как OCR, то большинство работ основаны на HMМ [3,4,5] или различных модификациях RNN [6,7]. Выше упомянутые методы позволяют принимать во внимание порядок символов. Если рассматривать задачу как OWR, то чаще всего используются сверточные нейронные сети** (Convolutional Neural Networks, CNN)[5]**. Здесь весь текст разделяется на слова, которые в дальнейшем распознаются по отдельности. В этом случае порядок слов не важен.**



**Рис. 6 Пример неоднозначного написания слова**

**Оба этих метода имеют свои преимущества и недостатки. К примеру, для OCR достаточно набрать датасет, содержащий буквы алфавита, а в случае с OWR необходимо будет создать огромный словарь целевых слов и собрать датасет, что значительно дороже, чем в предыдущем подходе. С другой стороны, OWR может быть точнее, так как, в отличие от OCR, у него не возникает проблем с разбиением слова на символы, что может быть проблематично в случае с русским или казахским языками (Рис. 2).**

В последнее время практически большинство методов распознавания используют глубокие нейронные сети, такие как сверточные нейронные сети (Convolutional Neural Networks, CNN) [5] и рекурентные нейронные сети (Recurrent Neural Networks, RNN) [6,7]. Более устаревшими решениями являются решения на базе скрытых Марковских моделей [3,4,5].

**2.1. СКРЫТЫЕ МАРКОВСКИЕ МОДЕЛИ (HMM)**

**В контексте HMM задачу можно сформулировать следующим образом: найти наиболее близкий набор слов при условии, что дан снимок строки.**

**Математически это можно выразить следующей формулой:**

**где *w* – это строка слов, *L* – язык, y – набор дескрипторов.**

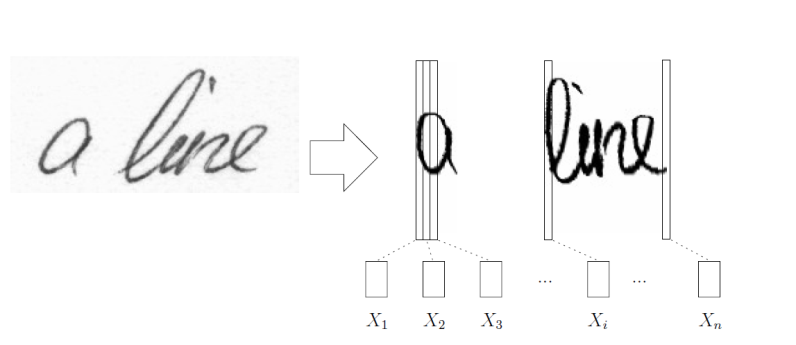
**С использованием формулы Байеса:**

**Для одной строки текста *y* и *P(y)* являются константами, следовательно, необходимо найти только следующее:**

**где *P(y|w)* – это модель правдоподобия, - состояния**

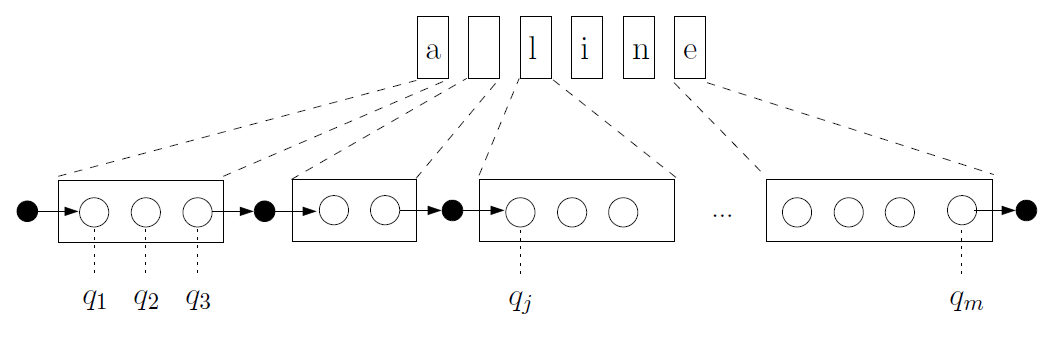
**а *P(w)* – априорная вероятность строки слов**

**Допустим, у нас есть строка, изображенная на Рис. 7. Во-первых, она нормализуется, во-вторых, строятся признаки строки .**



**Рис. 7 Пример строки**

**С учетом того, что имеется распределение состояний для строки (Рис. 8),**



**Рис. 8 Разбиение строки на состояния.**

**производится поиск нужных зависимостей между признаками и состояниями с использованием метода Viterbi (Рис. 9).**

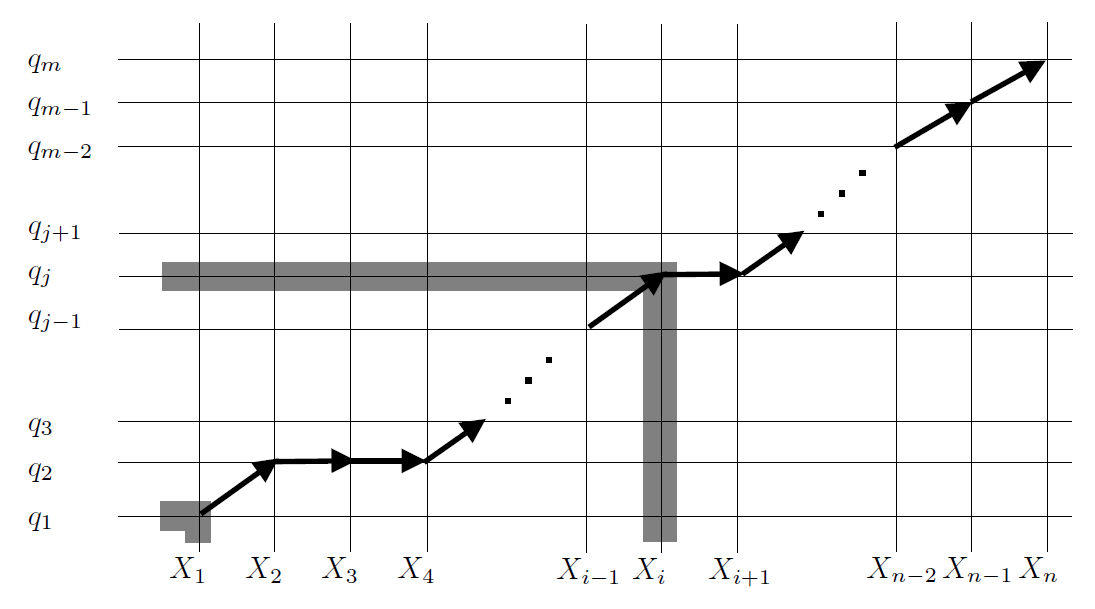


Рис. 9 Мэппинг дескрипторов с состояниями.

Основным недостатком этого метода является тот факт, что входные признаки должны удовлетворять условиям независимости. Другими словами, признаки должны быть изготовлены вручную для каждого алфавита.

**2.2. BIDIRECTIONAL LONG SHORT TERM MEMORY (BILSTM)**

Модель, распознающая только буквы кириллицы, простая и состоит из нескольких слоев сверточных нейронных сетей, слоев MaxPooling, и функции нелинейности ReLU. Далее идут несколько слоев Fully Connected Network, и в итоге softmax активация для подсчета распределения вероятностей. К сожалению, подобную логику данной модели тяжело использовать в случае распознавания слова целиком, так как вариаций слов слишком много, что предполагает наличие в выходном слое столько нейронов, сколько вообще целевых слов в датасете. В случае если мы рассматриваем применение модели вообще для любых слов, и нарицательных и собственных, то невозможно построить единый словарь наименований, так как чем больше классов в задаче обучения с учителем, тем меньше вероятность угадывания правильного ответа. Исходя из этого, для решения общей задачи распознавания текста нецелесообразно использовать подобный подход.

Интуиция решения вышеуказанной проблемы следующая: нужно последовательно обрабатывать буквы в слове. К сожалению, разделение слова на буквы не представляется возможным, в частности в русском языке часто слова пишутся слитно, и такие стандартные методы разделения слова на буквы как гистограммный метод не подходит. Поэтому не удастся найти место соединение слов. Следовательно, изображение слова нужно обрабатывать попиксельно. Мы использовали архитектуру модели описанную в статье [8], где авторы решают проблему распознавания арабского рукописного текста.

**2.2.1. Описание архитектуры**

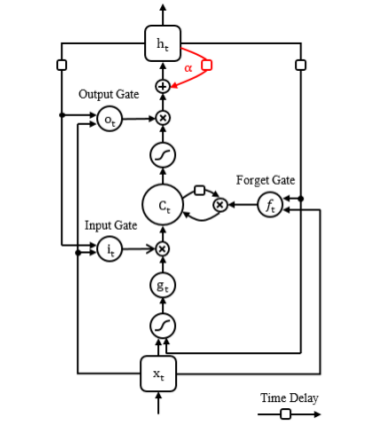
**Входные данные**

В качестве входных данных мы принимаем картинки высотой 60 пикселей и длиной 500 пикселей.

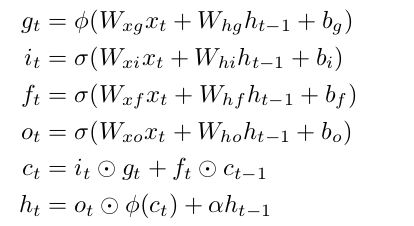
**Рекуррентные слои нейронной сети**

Как и в [8] мы использовали рекуррентную нейронную сеть состоящую из четырех слоев. Каждый слой состоит из ячейки Bidirectional Long Short Term Memory (BiLSTM). Все внутренние и скрытые вектора состояний каждого слоя инициализируются нулями. Первый и второй слой нейронной сети содержат по 100 внутренних нейронов. Входными данными для первого слоя является (в каждый момент времени) один столбец пикселей изображения (т.е. 60 пикселей). Так как мы используем BiLSTM ячейку для каждого направления данные начинают поступать из двух разных концов изображения. В каждый момент времени результатом текущего слоя является аффинное преобразование конкатенации двух векторов из разных направлений. Данная логика распространения информации сохраняется и в следующих слоях.

Далее, результаты первого слоя являются входными данными для второго слоя, который отрабатывает точно также как и первый слой. Затем выходные данные второго слоя конкатенируются с выходными данными первого слоя. Метод соединение выходов разных уровней называется Residual Skip Connection. Он помогает качественному распространению информации через слои глубоких нейронных сетей, и в случае излишества конкретного слоя, skip connection помогает сохранить информацию такой какой она была до входа в текущий слой. Далее полученный результат отправляется в следующий третий слой. В слоях 3 и 4 происходит то же самое что и с 1 и 2 слоями, с единственным различием - в 3 и 4 слоях количество нейронов равно 200. Конечным выводом этой части нейронной сети является вектор в каждый момент времени.

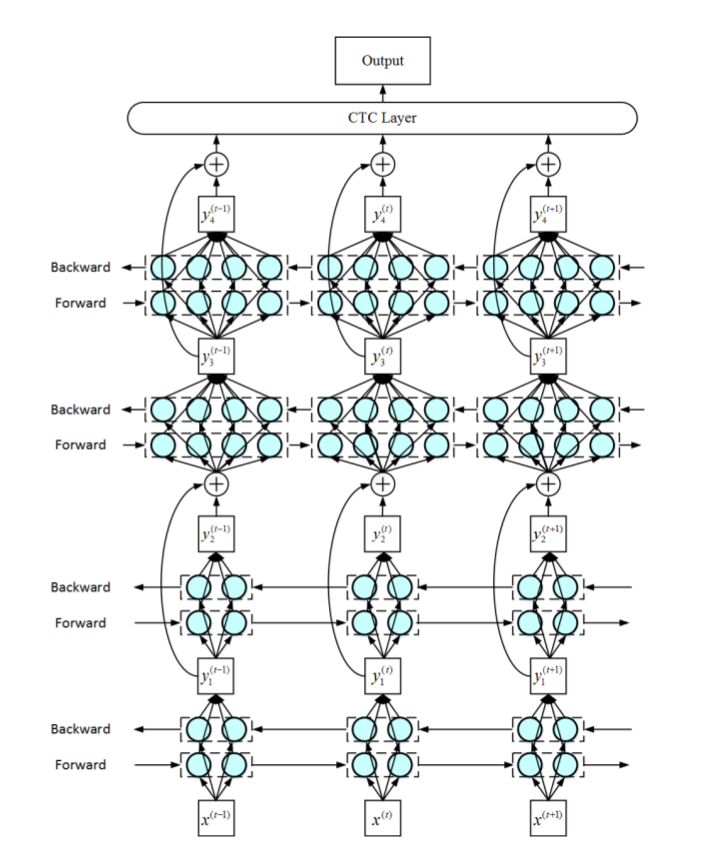


**Рис. 10** Описание LSTM ячейки



**Connectionist Temporal Classification (CTC) слой**

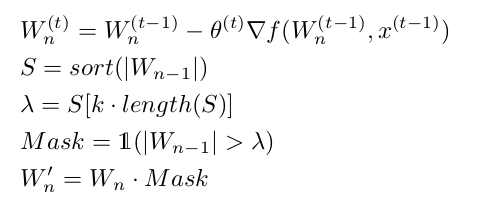
Итоговым слоем является Connectionist Temporal Classification, задача которого к входному вектору поставить в соответствие букву из алфавита или «пустой бланк», означающий отсутствие какой либо буквы в каждый момент времени. Для подробного описания см [9].

****

**Рис. 11** Общая нейронная сеть

**2.2.2. Этап обучения**

Подобная нейронная сеть обучается в два этапа: нормальный (normal) и разреженный (sparse). Сначала идет нормальный этап, когда нейронная сеть обучается как обычно - ошибки подсчитанные на последнем слое постепенно распространяются на нижние слои, меняя веса нейронов одним из разновидностей метода градиентного спуска, называемого методом Адама. На следующем этапе (разреженного обучения) веса нейронов, которые ниже предопределенного q-персентиля, приравниваются к нулю, тем самым распространяя ошибки на нейроны с большим весом.



**3. РАСПОЗНАВАНИЕ РУКОПИСНОГО АДРЕСА**

Любой язык содержит в себе огромное количество слов. К примеру, словари русского и казахского языков в среднем регистрируют более 100 000 слов, а оксвордский словарь английского языка – более 300 000 слов. В связи с этим собрать исчерпывающую базу данных рукописных слов, которая включала бы в себя все слова с большой вариацией почерка, представляется практически невозможным. Другими словами, всегда найдется слово, которое система не сможет распознать.

В отличие от общего случая с текстом, адрес может включать в себя слова из более ограниченного множества специфичных для адреса слов. В большинстве случаев, адрес - это короткий блок из 5-15 слов, в котором указываются страна, штат/провинция/город, улица, номер дома, почтовый индекс, фамилия и имя отправителя/получателя. Причем, фамилия и имя отправителя/получателя в решении задачи сортировки корреспонденции никаким образом не участвуют, т.к. не несут никакой информации о направлении письма, поэтому они могут быть опущены. Кроме того, порядок слов в адресе не имеет значения, так как любая перестановка элементов адреса не изменит конечного пункта назначения.

Дополнительно, в адресе наблюдается некая иерархия. К примеру, если в адресе в качестве страны указан Казахстан, то закономерно предположить, что далее последует один из населенных пунктов на территории РК. Таким образом пространство поиска может быть значительно сокращено, то есть нет необходимости вести поиск по всем существующим в мире городам, а можно ограничиться теми, которые находятся в определенном на предыдущем шаге регионе. Подобная иерархическая зависимость позволяет рассматривать более гибкие и легко масштабируемые архитектуры системы. Например, мы можем создать систему, которую «научим» распознавать название страны назначения. Согласно википедии, официально в мире насчитывается 197 стран. Далее, для каждой станы мы обучим отдельные независимые модели распознавания названий городов. Итак, по мере необходимости мы сможем расширять систему, причем количество необходимых моделей не будет превышать 197.

**3.1. ОСНОВНАЯ ИДЕЯ**

Основная идея нашего подхода заключается в разделении процесса распознавания адреса на несколько этапов.

Каждый этап действует на подмножестве слов, ограниченном предыдущим этапом. Другими словами, мы предлагаем следующую последовательность распознавания:

1. Распознавание страны
2. Распознавание второй по значимости административной единицы страны.
3. Распознавание доп. информации

Мы предлагаем реализовать пункты 1-2 с помощью по-словного распознавания (OWR), а пункт 3 -- с помощью по-символьного распознавания. Причем, граница перехода между word-based и char-based распознаваниями будет экспериментально подбираться. Например, названия улиц потенциально можно распознавать по-словно. Однако, если количество различных названий улиц в рамках одной страны окажется выше порогового уровня (временные, денежные и вычислительные затраты на сбор, обработку и хранение датасета), то более целесообразно будет распозновать их посимвольно.

Выделение страны в адресе позволяет определить генеральное направление, а определение верхнего административного пункта страны - уточнить генеральное направление. К примеру, в США к верхним адмиистративно-территориальным единицам можно отнести штаты (~50); для Китая – провинции и города центрального подчинения (~34); для Казахстана – области и города (~17). C нашей точки зрения, в контексте сортировки корреспонденции определение страны и административного пункта имеют наивысший приоритет и требуют максимально возможной точности.

Согласно информации с википедии, в мире насчитывается около 197 стран. Следовательно, чтобы обучить модель на определение страны нам достаточно собрать датасет из различных образцов написания 197 слов. Подобный набор слов является вполне «подъемной» задачей с точки зрения сбора образов почерков.

Учитывая тот факт, что наша задача тесно связана с деятельностью КазПочты, мы привязаны к корреспонденции, которая тем или иным способом проходит через Казахстан. Если продолжить размышления, то можно определить следующее. Для Казахстана все страны можно грубо разбить на ближнее зарубежье и дальнее зарубежье. К ближнему зарубежью мы будем относить те постсоветские страны, где до сих пор широко используется русский язык. К подобным странам относятся около 10 стран: Россия, Украина, Белоруссия, Казахстан, Узбекистан, Киргизстан, Таджикистан, Грузия, Армения, Азербайджан. Соответственно, дальнее зарубежье – страны, использующие английский язык в качестве международно принятого стандарта. В Казахстане к основным языкам, используемым при написании адресов, относятся казахский, русский и английский. Причем, при корреспонденции внутри страны чаще всего используется казахский и русский; при корреспонденции с ближним зарубежьем - русский; и с дальним – английский.

С учетом разных языков, есть смысл обучать две верхнеуровневых моделей – одну только для ближнего зарубежья (русский), вторую – для ближнего и дальнего зарубежья (английский). Первая модель будет обучаться на названиях 10 вышеупомянутых стран на кириллице. Вторая модель – на названиях всех стран на латинице.

**3.2. АРХИТЕКТУРА СИСТЕМЫ**

В качестве моделей мы планируем использовать сверточные нейронные сети (CNN).

1. Для казахского языка
   1. NN\_2\_1, NN\_2\_2, …, NN\_2\_252 – классификаторы второй по значимости административной единицы
   2. NN\_3\_1, NN\_3\_2, …, NN\_3\_252 – классификатор улиц
   3. NN\_4 – посимвольное распознавание
2. Для русского языка
   1. NN\_1 – классификатор для распознавания страны
   2. NN\_2\_1, NN\_2\_2, …, NN\_2\_252 – классификаторы второй по значимости административной единицы
   3. NN\_3\_1, NN\_3\_2, …, NN\_3\_252 – классификатор улиц
   4. NN\_4 – посимвольное распознавание
3. Для английского языка
   1. NN\_1 – классификатор для распознавания страны
   2. NN\_2\_1, NN\_2\_2, …, NN\_2\_252 – классификаторы второй по значимости административной единицы
   3. NN\_3\_1, NN\_3\_2, …, NN\_3\_252 – классификатор улиц
   4. NN\_4 – посимвольное распознавание

Здесь NN обозначает сверточную нейронную сеть, NN\_1 – модели верхнего уровня для распознавания названия страны назначения, NN\_2 – модели второго уровня (используется для распознавания административной единицы пункта назначения), NN\_3 – модели третьего уровня (для распознавания названия улиц), NN\_4 – модель для посимвольного распознавания доп. информации.

Преимущества данного подхода:

* Высокая точность определения генерального направления (страна --> штат/провинция/город --> улица)
* Интуитивное архитектурное деление системы
* Иерархичный подход, позволяющий значительно уменьшить пространство поиска
* Метод легко расширяем – для каждой страны (языка) мы можем тренировать классификатор по отдельности
* Метод позволяет работать с любыми типами конвертов

**3.3. ЭКСПЕРИМЕНТЫ**

Для первоначального прототипа мы планируем создать две модели. Первая модель будет определять страну. Соответственно, мы обучим ее на названиях стран ближнего зарубежья на русском языке (10 стран). Вторая модель будет распознавать области и города Казахстана на русском и казахском языках.

Данный эксперимент позволит нам проверить несколько ключевых моментов наших исследований, а именно:

* Объем данных, необходимых для обучения моделей
* Подобрать оптимальную архитектуру моделей
* Проверить предложенную архитектуру системы

В конце эксперимента мы предполагаем получить прототип работающей системы для сортировки корреспонденции по Казахстану. Далее, архитектура системы позволяет добавлять другие страны без внесения изменений в уже обученные модели.

Изначально мы делаем следующие допущения:

1. Нет переносов и ошибок
2. Нет паразитного текста
3. Слова написаны четко и хорошо видны
4. Нет перспективы

**4. СБОР ДАННЫХ**

Любая задача машинного обучения с учителем требует входных размеченных данных, на которых можно было бы обучить модель. Причем, нам необходимо обучить как минимум две модели: одну для определения областей снимка, где располагается текст, а другую для распознавания слов.

Следовательно, мы должны собрать как минимум два набора данных: снимки реальных конвертов, подобные тем, с которыми нам предстоит работать, а другой набор – множество рукописных целевых слов.

Мы предполагаем, что первый набор данных (снимки конвертов) мы получим от КазПочты. В данный момент ведутся интенсивные переговоры по поводу этого вопроса. Второй набор данных мы собираем собственноручно, так как подобного набора на казахском, русском языках нет в свободном доступе.

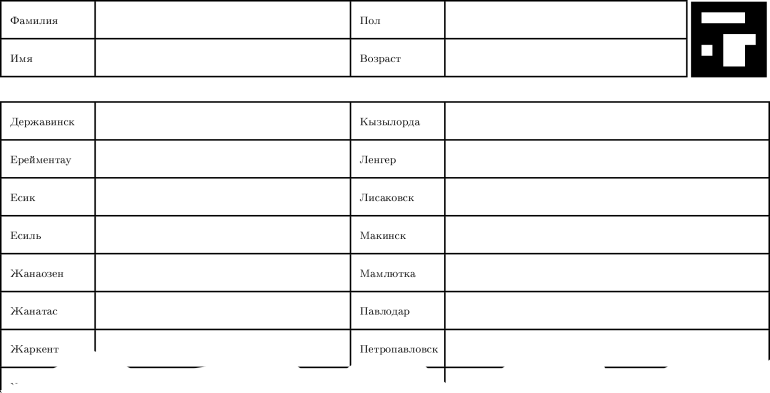


Рис 1. Пример формы для сбора слов

Для этих целей мы определили минимальный набор слов, который включает все названия городов, поселков, сел, районов и улиц Казахстана, и создали макеты форм для заполнения. Формы созданы таким образом, чтобы максимально упростить процесс «вырезания» слов из формы (Рис. 1).

Также были проведены исчерпывающие эксперименты, связанные с предобработкой форм с целью автоматической идентификации форм, определения контуров форм, компенсации вращения, а также удаления краевых артефактов на границах сегментированных слов.

На текущий момент проходят работы, связанные с нормализацией слов. В данном контексте, под нормализацией предполагается центрирование и масштабирование слов, определение и компенсация наклона почерка. После того, как слова будут собраны, они будут использованы для обучения выбранной модели распознавания текста.

## ****4.1. СУЩЕСТВУЮЩИЕ БАЗЫ ДАННЫХ РУКОПИСНОГО ТЕКСТА****

Существует множество датасетов с рукописным текстом. Большинство из них содержат текст на английском языке. Эти датасеты не ограничены контекстом, то есть генеральная популяция контекстов апроксимируется случайной выборкой. Одними из ярких примеров датасетов являются:

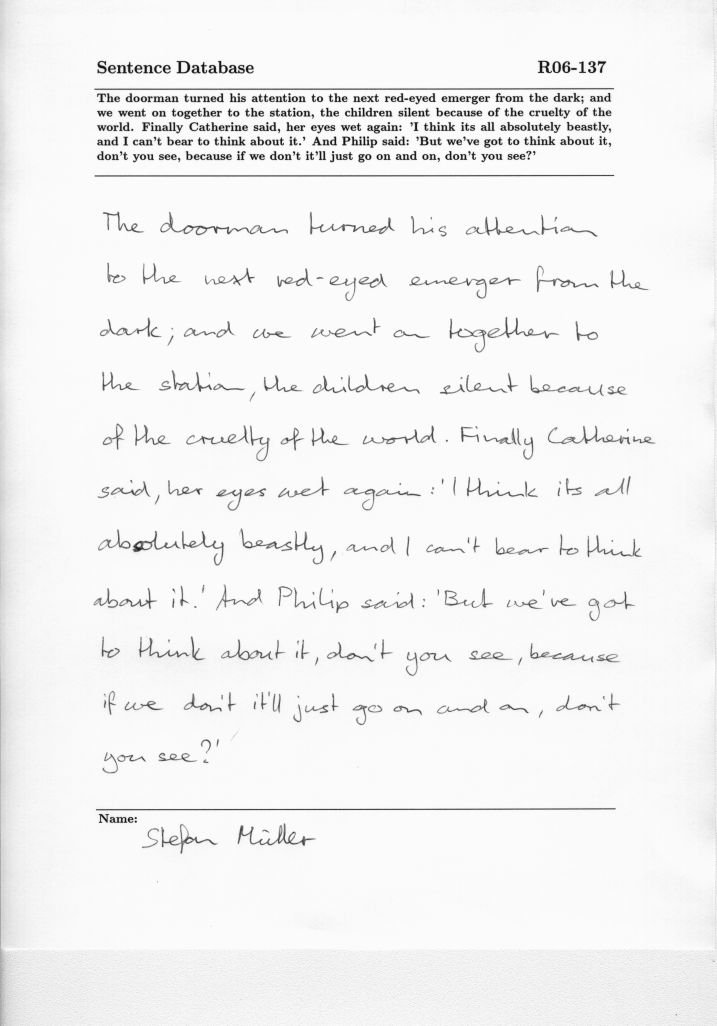
1. IAM Handwriting Database
2. CEDAR
3. **CVL-Database**
4. **ICDAR2013-2017**
5. NIST
6. CENPARMI
7. UNIPEN
8. IRESTE On/Off (IRONOFF) Dual Handwriting Database

Аналогов на русском и казахских языках не существует. Мы адаптировали общие принципы сбора и хранения данных, описанные в [Zimmermann] и использованные для сбора данных IAM Database. Описание данной базы данных смотрите ниже.

**4.1.1. ОПИСАНИЕ IAM DATABASE**

**Создатели данной базы данных использовали 13 категорий текста. В каждом тексте около 2000 слов, поделенных на 3-6 фрагментов по 50 слов. Каждая форма содержит 4 области, разделенные горизонтальными линиями:**

1. **Заголовок и идентификационный номер (категория-индекс текста-индекс предложения)**
2. **Печатный текст**
3. **Пустая зона для рукописного текста**
4. **Зона для имени (необязательно)**

****

**Рис 2. Образец формы из базы данных IAM.**

**Разделяющие линии используются для сегментации зон и определения пространственных деформаций листа при сканировании. После сегментации текст режится по линиям. У каждого слова в печатном тексте есть своя метка. Эти метки копируются для всего фрагмента, и словам в рукописном варианте, которые находятся в строке, в ручную присваиваются метки.**

**Результатом размечивания (labeling) является ASCII файл, который содержит ярлыки для каждой строчки печатного и рукописного текста. В файл не включается пространственная информация. Каждый участник опроса заполнял от одной до трех форм разных категорий.**

По данным с сайта известно, что в опросе учавствовали 657 человек. В общей сложности по шаблону, изображенному на Рис. 1, было собрано 1539 страниц текста. Эти страницы сегментированы на 5685 предложений, 13353 аннотированных строчек и 115 320 слов.

# ****4.1.2. ОПИСАНИЕ CVL-DATABASE****

**Количество людей, заполнивших формы, равно 310. База данных использовала 7 различных текстов (1 на немецком и 6 на английском). Документы сканировались с разрешением 2499x3520 пикселей. Каждой форме присвоены идентификационный номер заполнившего форму человека, а также номер текста.**

**Более подробная информация о том, как авторы собирали данные и сегментировали текст, отсутствует.**

# ****4.1.3. ОПИСАНИЕ ICDAR2013 DATASET****

ICDAR2013 использовалась в рамках конференции 2013 года. Она содержит 150 сканов рукописного текста на греческом и английском языках. Кроме того, 50 изображений содержат текст, написанный на бенгальском.

# ****4. 2. СБОР И ХРАНЕНИЕ ДАННЫХ****

Этап сбора данных является одним из наиболее трудоемких и затратных этапов. Наша основная задача – это максимально упростить и автоматизировать его. В контексте распознавания рукописного адреса необходимо определить множество ключевых слов, которые могут встречаються в адресе, и собрать следующее:

* образцы рукописного алфавита на кириллице и латинице

## **рукописные образцы ключевых слов** на казахском, русском, английском языках

## **сканы лицевой стороны конвертов**

Далее, необходимо будет сделать аннотацию собранных данных, т.е. разметить, и синтезировать новые образцы уже из существующих, посредством применения различных геометрических и фотометрических трансформаций (data augmentation).

## ****4.3. БАЗА ДАННЫХ КЛЮЧЕВЫХ СЛОВ****

Для начала, мы рассматриваем входящую в РК корреспонденцию, поэтому список ключевых слов содержит следующие названия:

* Областей
* Городов
* Сел
* Поселков
* Районов
* Улиц

Дополнительная информация такая, как:

* Индексы
* Телефоны
* Фамилии
* Названия компаний

в базу данных не входит.

## ****4.4.**** ОБРАЗЦЫ РУКОПИСНОГО АЛФАВИТА

Существует два принципиальных подхода распознавания текста: по символьное распознавание (Optical Character Recognition, OCR) и распознавание по словам (Optical Word Recognition, OWR).

При OCR для обучения модели требуется датасет, содержащий рукописные образцы всех символов алфавита для каждого языка. Здесь важно для каждого языка составить отдельные формы, так как набор букв различных алфавитов может сильно различаться.

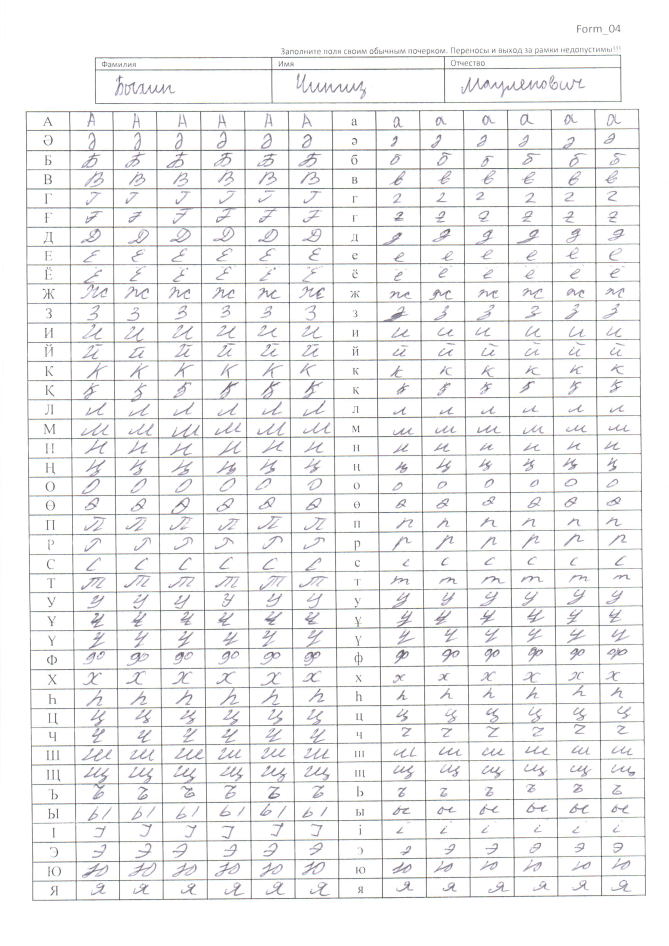


Рис 3. Один из первых вариантов формы для сбора рукописных образцов алфавита кириллицы.

Далее, для последующего обучения и тестирования модели понадобятся рукописные образцы целевых слов. Пример одной из форм для сбора образцов слов приведен на Рис. 3.

Данная форма спроектирована таким образом, чтобы была возможность легко идентифицировать и сегментировать по словам. Для идентификации формы в правом верхнем углу располагается маркер. Для упрощенния процесса сегментации вся форма разделена горизонтальными и вертикальными линиями, которые позволяют довольно легко восстановить структуру документа, и, соответственно, пространственное положение слова. Слова индексируются (аннотируются) согласно своего положения в таблице.

Для того чтобы вырезать слова из формы производятся следующие действия (preprocessing):

1. фильтрация форм с целью усиления границ таблицы
2. определение контуров таблицы
3. определение и компенсация угла поворота
4. исключение артефактных линий
5. сортировка форм по id (маркеру)
6. потоковое разделение форм на слова
7. наименование и хранение слов

После того, как области изображения, соответствующие ячейкам со словами сегментированы, они могут содержать некоторые краевые артефакты. Например, к артефактам можно отнести линии таблицы, вырезанные вместе с ячейкой, или части слова из соседней ячейки (Рис. 4). Мы устраняем эти артифакты, строя вертикальную и горизонтальные гистограммы и отсекая части отделенно локализированные ближе к краям ячейки.

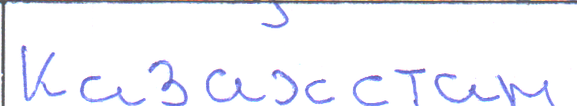


Рис 4. Пример вырезанной из формы области со словом. По краям видны ярко выраженная линия ячейки и кусочек буквы из соседней области.

Однако, не всегда возможно устранить все артефакты. Ниже приведены некоторые аспекты, затрудняющие дальнейшую обработку сегментированного слова:

1. буквы могут быть не соединены между собой
2. буквы могут персекаться с артефактами
3. положение букв и их крупность значительно варьирируются от слова к слову
4. буквы могут быть написаны разным цветом (синим, черным, красным)

В связи с этим, мы разработали систему рекоммендаций, которая позволяет упростить процесс выделения областей со словами из формы.

Во-первых, мы предлагаем заполнять форму с использованием синей ручки**.** Это позволит системе отличать слово от границ таблицы на уровне цвета. Например, переведя изображение из RGB в HSV, мы получаем инвариантное по отношению к освещению цветовое представление объектов. В данном цветовом пространстве синие остается синим вне зависимости от яркости и интенсивности изображения.

Во-вторых, устранение частей слов из соседних ячеек порой невозможно без искажения целевого содержимого данной ячейки, поэтому при заполнении формы желательно, чтобы испытуемый не выходил за границы ячейки.

Далее можно произвести нормализацию слов, то есть:

* 1. привести все изображения слов к одному размеру
  2. центрировать слово на изображении
  3. уменьшить размер изображения

Однако, современные модели, такие как для конволюционных и рекуррентных нейронных сетей данный этап может быть опциальным, так как их тренировка предполагает данные, содержащие любые геометрические трансформации объектов, изображенных на картинках.

**4.5. СБОР СКАНОВ ЛИЦЕВЫХ СТОРОН КОНВЕРТОВ**

После детального анализа задачи было обнаружено, что мы можем столкнуться со следующими сложностями при разработке ядра системы [2]:

1. Расположение текста на конверте произвольное
2. Формы, размеры и цвета конвертов сильно варьируются
3. **На конвертах могут присутствовать рисунки**
4. **Может присутствовать паразитный текст или печати**
5. **Наличие марок с текстом на конвертах**
6. **Печати могут быть поставлены поверх текста**
7. **Неортогональное расположение камеры относительно лицевой стороны конверта**
8. Ошибки в тексте (с/без исправлений)
9. **Сокращения в тексте**
10. **Перенос слов**
11. **и т.д.**

Следовательно, задача является нетривиальной. Для точного выделения областей с адресами необходимо большое количество изображений конвертов, наиболее часто встречающихся в обозначенном регионе, для того чтобы обучить модель. Если мы хотим чтобы наша система работала с большой точностью, то эти изображения должны в достаточной мере содержать все возможные вариации по форме, содержанию и различным паразитным артефактам.



Рис 5. Снимки конвертов под разным углом.

Каждый конверт содержит свой адрес

## ****4.6. СПОСОБЫ СБОРА ДАННЫХ****

**Человек, который согласился предоставить образец своего почерка, должен будет сделать следующее:**

1. **Получить целевой текст.**
2. **Написать его на чистом листе A4.**
3. **Сфтографировать.**
4. **Выслать нам.**

**Это можно организовать следующим образом:**

* **email**
* **разработка и запуск telegram-бота**
* **установка оборудования для сканирования адресов в пунктах приема писем**
* **Сканирование старой имеющейся корреспонденции**

**5 ЗАКЛЮЧЕНИЕ**

В течение года мы тщательно изучили вопрос распознавания рукописного текста; определили основные сообщества и конференции, рассматривающие этот вопрос; изучили наиболее широко используемые современные методы решения данной проблемы; выявили основные отличительные черты задачи распознавания адреса от распознавания текста; а также выдвинули концептуально новый способ решения задачи распознавания рукописного адреса на основе специфики последнего, что позволило нам создать довольно гибкую, интуитивно понятную и легко масштабируемую архитектуру системы.

Помимо этого, нами были спроектированы и запущены формы для сбора образцов почерка по ключевым словам. Набор ключевых слов мы также составили на основе наших экспериментов и наблюдений. Они используются для полученния данных для обучения моделей.

В текущее время работы по выбору оптимальной архитектуры моделей и сбору данных продолжаются согласно календарному плану.

# ****5. ЛИТЕРАТУРА****

1. Sargur. N. Srihari, Edward. J. Kuebert, Integration of Hand-Written Address Interpretation Technology into the United States Postal Service Remote Computer Reader System, 1997.
2. Alessandro Vinciarelli, A survey on off-line Cursive Word Recognition, Pattern Recognition 35 (2002) 1433–1446.
3. U.-V. Marti and H. Bunke. Towards general cursive script recognition. In S.-W. Lee, editor, Advances in Handwriting Recognition, pages 203–212. World Scientific, 1999.
4. Zimmermann, Bunke, Automatic Segmentation of the IAM Off-line Database for Handwritten English Text, 2002.
5. Jaderberg, Simonyan, Vedaldi, Zisserman, Reading Text in the Wild with Convolutional Neural Networks, 2014.
6. **A. Graves, S. Fernandez, and J. Schmidhuber. Multidimensional recurrent neural networks. In Proceedings of the 2007 International Conference on Artificial Neural Networks, Porto, Portugal, September 2007.**
7. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, November 1998.
8. Residual Recurrent Neural Network with Sparse Training for Offline Arabic Handwriting Recognition
9. Connectionist temporal classification: labeling unsegmented sequence data with recurrent neural networks
10. Sargur N. Srihari, Sung-Hyuk Cha, Hina Arora, Sangjik Lee, Individuality of Handwriting, J Forensic Sci, July 2002, Vol. 47, No. 4.