# **ВВЕДЕНИЕ**

1. ЭТАПЫ РАБОТ
   1. СБОР ДАННЫХ
   2. РАСПОЗНАВАНИЕ
2. ОБЗОР МОДЕЛЕЙ РАСПОЗНОВАНИЯ
   1. СКРЫТЫЕ МАРКОВСКИЕ МОДЕЛИ (HMM)
   2. РЕКУРЕНТНЫЕ НЕЙРОННЫЕ СЕТИ (MULTIDIMENSIONAL LSTM)
3. ЗАКЛЮЧЕНИЕ
4. 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.

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.**

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 рекуррентных нейронных сетях []. В главе 4 мы опишем выбранную нами модель и приведем промежуточные результаты. Глава 5 содержит заключение.

## ****2 ЭТАПЫ РАБОТ****

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.

**2.1. 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.

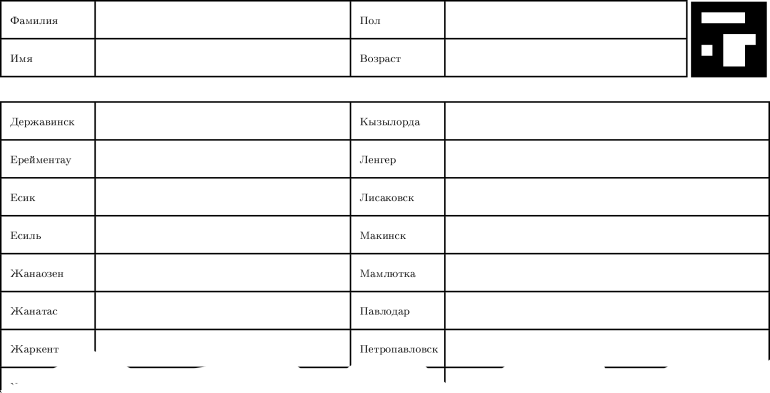


Рисунок 1. Пример формы для сбора слов

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.

**2.2. РАСПОЗНОВАНИЕ**

**После того, как данные для обучения модели распознавания рукописных слов собраны необходимо выбрать подходящие дескрипторы для получения ключевых признаков. Выбор дескриптора сильно зависит от используемой модели и выбора метода решения. 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 RNN [6,7]. Эти методы позволяют принимать во внимание порядок символов. Однако мы предполагаем, что в случае, если рассматривать задачу как OWR достаточно будет обычной нейронной сети, так как порядок слов неважен при определении почтового адреса. Оба этих метода имеют свои преимущества и недостатки.**



**Рис. 2 Пример неоднозначного написания слова**

**К примеру, 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 и казахским языком (Рис. 2).**

**3. ОБЗОР МОДЕЛЕЙ РАСПОЗНОВАНИЯ**

Задача распознавания рукописного текста представляется сложным. В последнее время практически все методы 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 моделях [3,4,5].

**3.1. 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.**

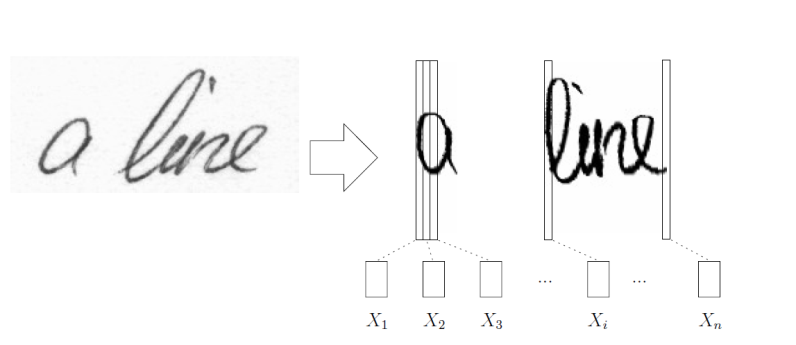
**С использованием формулы Байеса:**

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

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

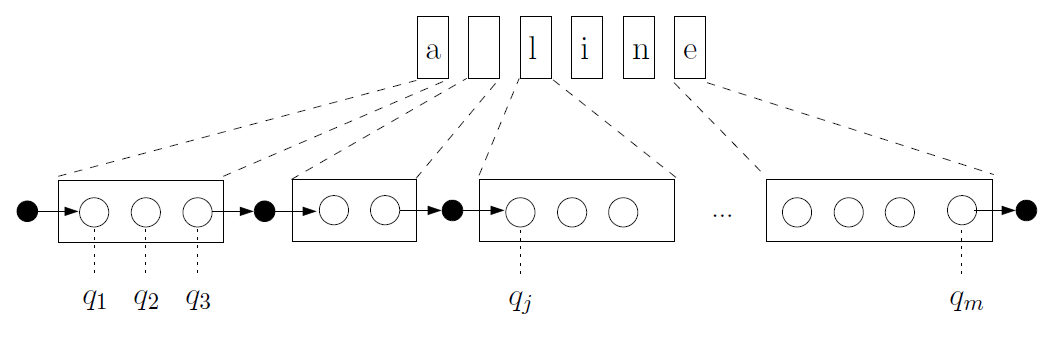
**а *P(w)* – априорная вероятность строки слов**

**Допустим, у нас есть строка, изображенная на Рис. 3. Во-первых, она нормализуется, во-вторых, строятся признаки строки .**



**Рис. 3 Пример строки**

**С учетом того, что имеется распределение состояний для строки (Рис. 4),**



**Рис. 4 Разбиение строки на состояния.**

**производится поиск нужных зависимостей между признаками и состояниями с использованием метода Viterbi (Рис. 5).**

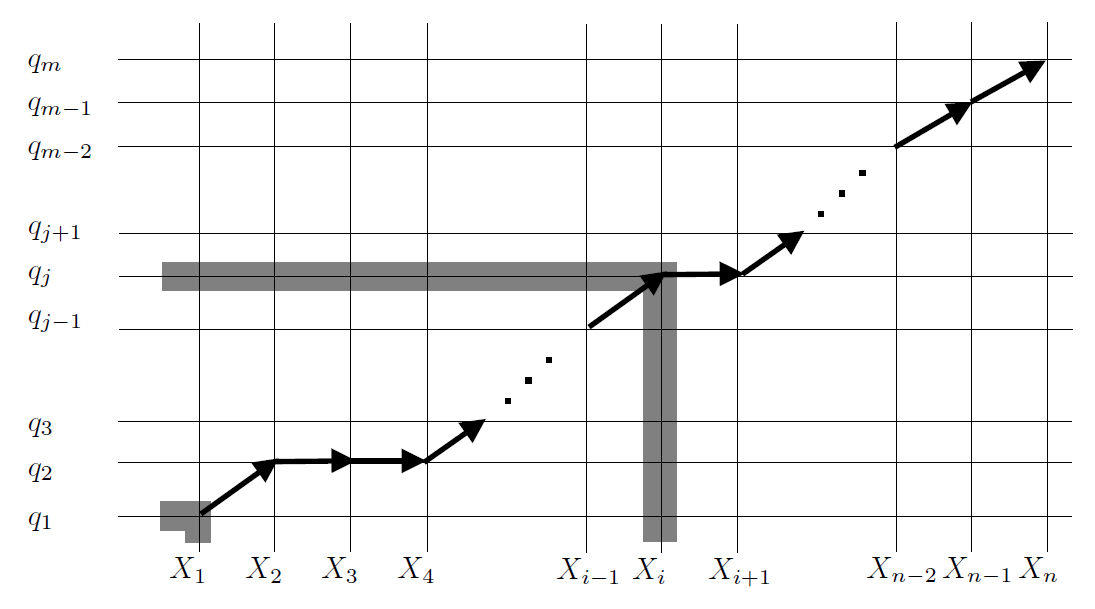


Рис. 5 Мэппинг дескрипторов с состояниями.

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

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

Построение модели распознающей 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, данные логику этой модели тяжело использовать в случае распознавания слова целиком, так как вариаций слишком много. В таком случае в выходном слое должны быть столько нейронов сколько вообще слов возможных в датасете. Так как мы рассматриваем применение модели вообще для любых слов, и нарицательных и собственных, невозможно построить единый словарь наименований. Сложности очевидно можно увидеть в случае когда мы хотим распознавать имена или название городов. Мы знаем что, чем больше классов в задаче обучения с учителем тем меньше вероятность угадывания правильного ответа. Поэтому нужно отказаться от классификации слова сразу целиком.

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, картину слова нужно обрабатывать попиксельно. Мы использовали архитектуру модели описанную в статье [8], где авторы решали проблему распознавания арабского рукописного текста.

**3.2.1. Описание архитектуры**

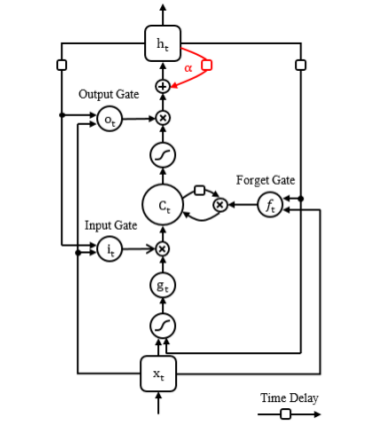
**Входные данные**

Входными данными мы понимаем картинки одинаковой высоты в 60 пикселей и длины 500 пикселей. Слова, которые слишком короткие и не доходят по длине до 500 пикселей мы заполняем недостающую длину нулями. Слова, которые по высоте отличающееся от 60 пикселей растягиваем, либо сжимаем.

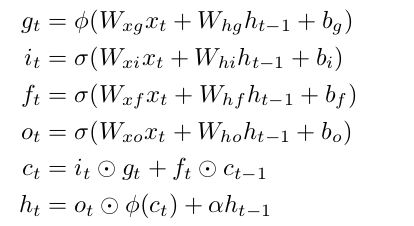
**Рекуррентные слои нейронной сети**

Как и в [8] мы использовали рекуррентную нейронная 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. Первой и второй слой нейронной сети содержат по 100 внутренних нейронов. Входными данными для первого слоя является в каждый момент времени один столбец пикселей из картины высотой в 60 пикселей. Так как мы используем BiLSTM ячейку для каждого направления данные начинают поступать из двух разных концов картины слова размером 60х500. 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 входных данным 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 по 200. Итогом выводом этой части нейронной сети является вектор в каждый момент времени.

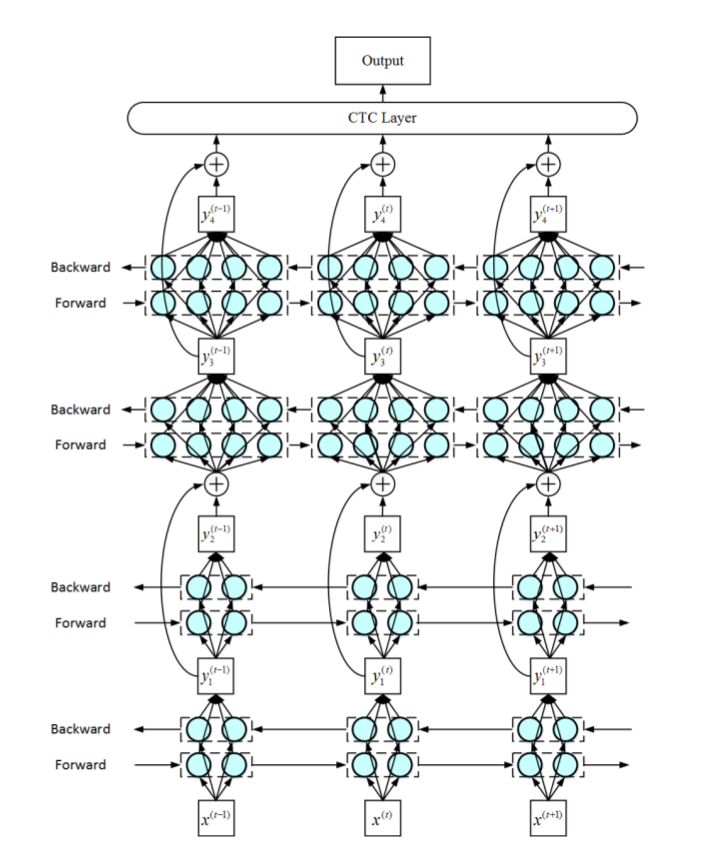


**Рис. 6** Описание LSTM ячейки



**Connectionist Temporal Classification (CTC) слой**

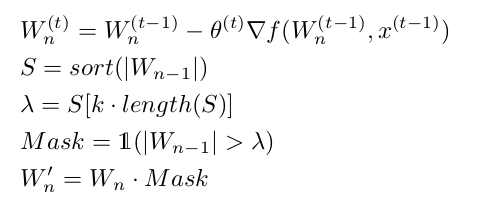
Итоговым слоем является Connectionist Temporal Classification, задача которого к входному вектору поставить в соответствие букву из алфавита или пустой бланк, означающий отсутствие какой либо буквы в каждый момент времени. Для подробного описания см [9].

****

**Рис. 7** Общая нейронная сеть

**3.2.2. Этап обучения**

Наша нейронная сеть обучается в двух этапов: нормальный(normal) и редкий(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 разновидностей метода градиентного спуска называемый метод Адама. В следующем этапе редкого обучение веса нейронов, которые ниже наперед заданного q-percentile ,приравниваются к нулю, тем самым распространяя ошибки на нейроны с большим весом



**4 ЗАКЛЮЧЕНИЕ**

Задача распознавания рукописного текста представляется сложным. В последнее время практически все методы распознавания используют глубокие нейронные сети, такие как сверточные нейронные сети и рекурентные нейронные сети. В данной работе мы ставим цель распознать английский рукописный текст, а именно картинки из базы данных IAM Handwriting Database.

Нами была разработана модель распознавания целого рукописного слова. В дальнейшем мы планируем опробовать модель на более сложном датасете слов на кириллице. Сложность заключается в том что, рукописные слова в основном пишутся слитно, т.е. представляет сложность разделения слова на буквы.

# ****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.