МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ К.И. САТПАЕВА

УДК 004.41, 004.931, 004.8

МРНТИ 20.15.13, 20.19.27, 20.19.29, 28.23.37

№ гос.рег. 0118РК00284

Инв. №

УТВЕРЖДАЮ

Проректор по науке

д-р техн. наук, проф.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Б.К. Кенжалиев

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

ОТЧЕТ

О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

РАЗРАБОТКА СОБСТВЕННОЙ МОДЕЛИ ОПТИЧЕСКОГО РАСПОЗНАВАНИЯ РУКОПИСНОГО ТЕКСТА И ЕГО ОТЛАДКА

по теме:

Разработка и внедрение системы по распознанию рукописных адресов письменной корреспонденции АО «Казпочта» с использованием машинного обучения

(промежуточный, № АР05135175)

Грантовое финансирование научных исследований на 2018-2020 гг.

Научный руководитель проекта

к.ф.-м.н. Д. Нурсеитов

Алматы 2019

Список исполнителеЙ

|  |  |  |
| --- | --- | --- |
| Научный руководитель,  главный научный сотрудник, канд. физ.-мат. наук | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | Нурсеитов Д.Б. |
|  | подпись, дата |  |
| Исполнители: |  |  |
| Ведущий научный сотрудник, доктор PhD | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | Бостанбеков К.А. |
|  | подпись, дата |  |
| Старший научный сотрудник, доктор PhD | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | Алимова А.Н. |
|  | подпись, дата |  |
| Старший научный сотрудник, докторант PhD | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | Курманходжаев Д.Б |
|  | подпись, дата |  |
| Научный сотрудник, докторант PhD | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | Канатов М. |
|  | подпись, дата |  |
| Научный сотрудник | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | Сагимбаев Ж. |
|  | подпись, дата |  |
| Нормоконтролер | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | Алимова А.Н. |
|  | подпись, дата |  |

РЕФЕРАТ

Отчет содержит: 28 стр., 20 рис., 6 библиограф. источников, 2 приложений.

OCR, OWR, ОПТИЧЕСКОЕ РАСПОЗНАВАНИЕ, НЕЙРОННЫЕ СЕТИ, РУКОПИСНЫЙ ТЕКСТ, МАШИННОЕ ОБУЧЕНИЕ.

*Объект исследования или разработки.* Распознавание рукописного текста на адресном поле письменной корреспонденции.

*Цель работы.* Заключается в разработке инновационного программного обеспечения с использованием современных методов машинного обучения на базе нейронных сетей для решения задачи распознавания и обработки естественного языка (natural language processing), состоящей в оптическом распознавании символов рукописного текста на русском и казахском языках.

*Название этапа работ на 2019 год.* Разработка собственной модели оптического распознавания рукописного текста и его отладка

*Метод или методология проведения работы.* В рамках данного исследования, для решения проблемы распознавания рукописного текста, будут применяется методы машинного обучения, такие как рекуррентные и сверточные нейронные сети.

*Результаты работы и их новизна.* Были изучены основные разработки в сфере распознавания рукописного текста для почтовой корреспонденции. Они направлены в основном на решение проблем определения интересующей области, сегментации текста, избавления от мешающих работе с текстом фоновых шумов, таких как утраченные или неясные фрагменты, пятна на бумаге, обнаружение перекоса, а также на обучение искусственного интеллекта, для распознавания написанного текста на интересующем нас языке. Проанализированы наиболее часто используемые в данном контексте модели распознавания, а именно модели на основе скрытых марковских моделей (HMM), гибридных марковских моделей (Hybrid HMM), сверточных (CNN) и рекуррентных нейронных сетях (RNN).

Любая задача машинного обучения с учителем требует входных размеченных данных, на которых можно было бы обучить модель. В нашем случае необходимо обучить как минимум две модели: одну для определения областей снимка, где располагается текст, а другую для распознавания слов. Были спроектированы и запущены формы для сбора образцов почерка по ключевым словам. Для казахского языка был проведен сбор рукописного текста букв алфавита (480 вариантов на каждую букву), включая казахские буквы, и наименовании городов по специально подготовленной нами форме (порядка 200 слов по 80 вариантов на каждое слово). Также сформирован набор данных из сканированных изображений лицевых сторон конвертов с рукописным текстом, для обучения определения интересующей нами областей снимка. Была обучена модель для обнаружения области с рукописным текстам на лицевой стороне конверта. Реализован алгоритм сегментации обнаруженного текстового блока по строкам и словам с помощью построения гистограмм.

Выявлены основные отличительные черты задачи распознавания адреса от распознавания текста, которые породили 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.

*Основные конструктивные, технологические и технико-эксплуатационные характеристики.* Автоматизированная работа центров обработки почтовых отправлений с увеличенным уровнем сервиса и скоростью доставки почты внутри Республики Казахстан.

*Область применения.* Реализация данного проекта позволит разработать систему автоматизированной обработки почтовой корреспонденции, включая рукописные адреса, максимально приближенной к бизнес-процессам почтовых служб.

*Экономическая эффективность или значимость работы.* Экономический эффект внедрения результатов данного проекта заключается в сокращении издержек при сортировке письменной корреспонденции в почтовых службах.

*Прогнозные предложения о развитии объекта исследования (НИОКР, бизнес-инкубирование, коммерциализация и т.д).* Результаты, полученные в рамках данного исследования, позволит развивать новые проекты, построенные на базе теории распознавания образов с помощью методов машинного обучения. Помимо распознавания адресов, модель распознавания рукописного текста может быть применена и для других задач, связанных с обработкой естественного языка. Имеется потенциал внедрения результатов проекта в частные почтовые службы.

# ВВЕДЕНИЕ

Основной целью данной работы является разработка системы по распознанию рукописных адресов письменной корреспонденции с использованием машинного обучения. Если описывать процесс работы конечной системы, то он должен 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.

Наше исследование не предусматривает разработку аппаратной части конечной системы. Основная цель нашего исследования ориентирована на разработке программной составляющей системы по распознанию рукописных адресов.

В первый год реализации проекта мы изучили существующие решения и определили наиболее эффективные на сегодняшний день подходы. Были собраны написанные от руки наименования стран и городов с различными образцами почерков, и был сформирован набор данных из сканированных изображений лицевых сторон конвертов с рукописным текстом.

Задачей на 2019 год является разработка собственной модели оптического распознавания рукописного текста. Для ее решения требуется разработка системы, модули которых зависят друг от друга. В соответствии первой половине календарного плана мы провели опыты с различными методами машинного обучения и разработали модули разрабатываемой системы. Данная система может быть разделена на следующие модули:

1. *Модуль предобработки входных картинок* (устранение шума, выравнивание по горизонтали). В данный модуль в качестве входных данных подается необработанная картинка конверта. В этом модуле устраняются шумы, а также оценивается угол поворота объекта вокруг оси, перпендикулярной плоскости конверта. После чего картинка выравнивается, то есть конверт поворачивается таким образом, что весь текст становится горизонтальным. Отчищенная от шума, и повернутая картинка является конечным результатом работы данного модуля.
2. *Модуль определения региона с текстом.* На лицевой части конверта, независимо от страны присутствуют два основных поля – это поле отправителя и получателя. Задачей данного модуля является определение этих регионов. Задача является весьма нетривиальной, и ее сложно решить в общем случае, так как, на практике, помимо указанных выше полей, на конверте могут присутствовать всякого рода «паразитные» объекты. Например, марки, рисунки, пометки, печати и логотипы компаний. Они сильно затрудняют задачу определения целевых регионов. В идеальном случае данный модель должен получать на вход выровненную картинку конверта и выдавать координаты двух регионов, соответствующих полям отправителя и получателя.
3. *Модуль сегментации региона с текстом по словам.* На данном этапе работы системы предполагается, что правильно найдены координаты регионов отправителя и получателя. Так как нашей конечной задачей является сортировка письменной корреспонденции по получателю, то нас конкретно интересует поле получателя. Задача данного модуля заключается в сегментации многострочной области по линиям, а затем, по словам. Результатом работы модуля сегментации должен быть набор координат, соответствующий регионам слов.
4. *Модуль распознавания слов.* После того, как сегментация картинки на отдельные регионы слов прошла успешно, можно начать непосредственное распознавание слов. За это ответственен модуль распознавания рукописных слов. Работа данного модуля базируется на рекуррентной нейронной сети со специального рода декодером, позволяющим более точно определять порядок и наличие символов на изображении. Данная модель проводит посимвольное распознавание. На вход модели подается вырезанная с конверта картинка слова, а на выходе выдается слово.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc11347929)

[1 ОПЫТЫ С СУЩЕСТВУЮЩИМИ РЕШЕНИЯМИ НА СОБРАННОМ ДАТАСЕТЕ И СРАВНЕНИЕ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ ДЛЯ РАЗНЫХ РЕШЕНИЙ 8](#_Toc11347930)

[1.1 Детектирование рукописных слов на конверте 8](#_Toc11347931)

[1.2 Подготовка тренировочных данных для модуля распознавания слов 12](#_Toc11347932)

[1.2.1 Первоначальная обработка картинок 13](#_Toc11347933)

[1.2.2 Генерация синтетических данных 14](#_Toc11347934)

[2 ПОСТРОЕНИЕ НЕСКОЛЬКИХ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧИ И ПРОВЕДЕНИЕ ОПЫТОВ С КАЖДЫМ РЕШЕНИЕМ 17](#_Toc11347935)

[2.1 Классификация рукописных слов с использованием Deep Convolutional Neural Networks 17](#_Toc11347936)

[2.2 Классификация рукописных слов с использованием модели SimpleHTR 20](#_Toc11347937)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc11347938)

[ПРИЛОЖЕНИЕ А Календарный план 25](#_Toc11347939)

[ПРИЛОЖЕНИЕ Б Список опубликованных работ 28](#_Toc11347940)

# 1 ОПЫТЫ С СУЩЕСТВУЮЩИМИ РЕШЕНИЯМИ НА СОБРАННОМ ДАТАСЕТЕ И СРАВНЕНИЕ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ ДЛЯ РАЗНЫХ РЕШЕНИЙ

## 1.1 Детектирование рукописных слов на конверте

Первой задачей при распознавании адреса на конверте является нахождение региона слова либо последовательность слов. Для этого необходимо обучить программу понимать изображение, используя методы компьютерного зрения (CV - Computer Vision).

Классификация изображений. Самой фундаментальной проблемой области CV является классификация изображений, где дается изображение, и мы ожидаем, что программа выдаст дискретный ответ. При этом мы предполагаем, что изображение содержит только один объект (Рисунок 1).

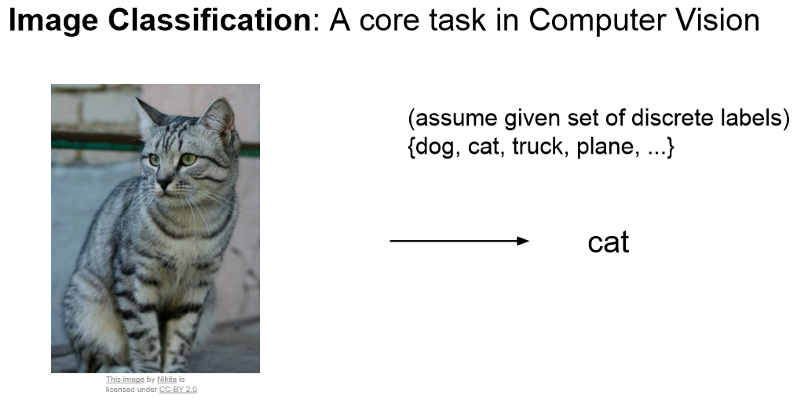


Рисунок 1 – Классификация изображений

Классификация с локализацией. В задаче локализации мы также производим классификацию объекта, но также и находим где точно объект находиться на изображении. Эта локализация обычно достигается с помощью Bounding Box, который определяется числовыми параметрами области где находится объект. Даже в этом случае мы ожидаем только один объект на изображении (Рисунок 2).

Детекция объектов расширяет задачу локализацию, где изображение не обязательно содержит только один объект, но множество и не обязательно одного класса. Здесь также применяется понятие Bounding Box (Рисунок 3).

Семантическая сегментация. Ее целью является классифицировать каждый пиксель изображения. Тут уже входными данными для обучения являются само изображение и маска для этого изображения (Рисунок 4).



Рисунок 2 – Классификация и локализация



Рисунок 3 – Детекция объектов

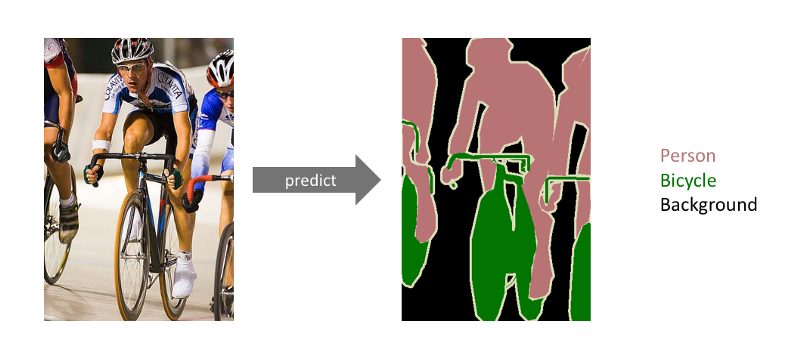


Рисунок 4 – Сегментация объектов

Сегментация с классификацией. Здесь помимо классификации каждого пикселя, производится классификация самого объекта на маске (Рисунок 5).



Рисунок 5 – Сегментация объектов с классификацией

Для задачи нахождения рукописных слов на конверте нам достаточно локализации объектов на самом этом изображении с конвертом. В предыдущем этапе исследований были собраны изображения конвертов, и после сортировки и чистки изображений было получено 400 изображений (Рисунок 6).



Рисунок 6 – Пример изображения конверта

Далее осуществлялась разметка данных, для этого каждое рукописное слово было выделено прямоугольником (Рисунок 7). Для разметки использовалась специальная программа Bbox Label Tool, написанная на языке Python.

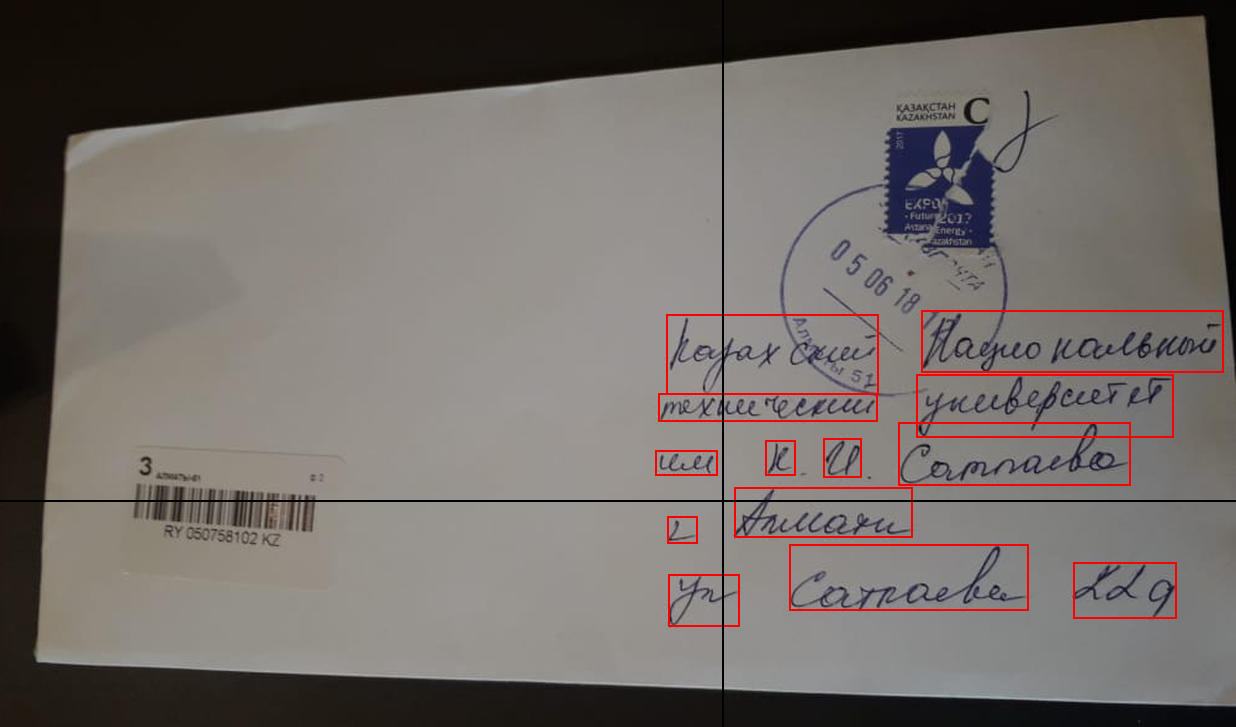


Рисунок 7 – Пример разметки изображений

Модель. В качестве модели для обучения изначально было выбрано два подхода:

* Mask R-CNN
* YOLO [1]

Mask R-CNN развивает архитектуру Faster R-CNN путём добавления ещё одной ветки, которая предсказывает положение маски, покрывающей найденный объект, и, таким образом решает уже задачу сегментации объектов (instance segmentation) (Рисунок 8). Маска представляет собой просто прямоугольную матрицу, в которой 1 на некоторой позиции означает принадлежность соответствующего пикселя объекту заданного класса, 0 – что пиксель объекту не принадлежит.

Авторы документа условно разделяют разработанную архитектуру на CNN-сеть вычисления признаков изображения, называемую ими backbone, и head ­– объединение частей, отвечающих за предсказание охватывающей рамки, классификацию объекта и определение его маски.

Выделение маски происходит в class-agnostic стиле: маски предсказываются отдельно для каждого класса, без предварительного знания, что изображено в регионе, и потом просто выбирается маска класса, победившего в независимом классификаторе. Утверждается, что такой подход более эффективен, чем опирающийся на априорное знание класса.

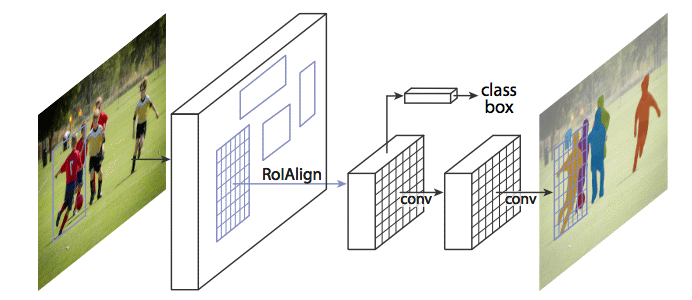


Рисунок 8 – Сегментация объектов на изображении

YOLO – это передовая система обнаружения объектов в реальном времени. На официальном сайте вы можете найти SSD300, SSD500, YOLOv2 и Tiny YOLO, которые прошли обучение с двумя различными наборами данных: VOC 2007+2012 и COCO. Еще больше вариантов конфигураций и наборов данных для машинного обучения вы можете найти в Интернете (например, YOLO9k). Благодаря широкому диапазону доступных вариантов можно выбрать версию, наиболее подходящую для ваших нужд. Например, Tiny YOLO – это самый «компактный» вариант, который может работать быстро даже на смартфонах или Raspberry Pi.

Так как для обучения модели Mask-RCNN необходимо было производить сегментацию, что является избыточной задачей для текущего проекта, была выбрана модель YOLO.

## 1.2 Подготовка тренировочных данных для модуля распознавания слов

Данный модуль использует модель на основе нейронных сетей. Для тренировки и тестирования данной модели должен быть подготовлен достаточно большой и репрезентативный набор данных. Данные, то есть рукописные слова, вырезанные из форм, которые мы собрали на предыдущем этапе исследований, в чистом виде, не являются достаточными, так как они содержат всего 500 образцов написания каждого слова из набора целевых слов (всего ключевых слов 42). Для достижения высокой общей точности модели необходимо увеличить набор данных хотя бы до 5000 образцов для каждого слова. Для этого мы решили генерировать новые изображения из имеющихся. Данная техника в западных источниках называется техникой дополнения картинок (image augmentation). Она позволяет генерировать новые картинки, внося некоторые изменения в исходную картинку. К примеру, к таким вариациям можно отнести перемещение, вращение, изменение наклона слова, а также добавление шума, кадрирование, изменение крупности слова и т.д. Дополнительно, для имитации разных почерков используется техника составления слов из набора рукописных символов. Кириллические рукописные символы были собраны нами параллельно с набором рукописных слов. На текущий момент у нас имеется около 80 форм рукописных заглавных и строчных символов кириллицы, где каждый символ повторяется по шесть раз.

Ниже более детально описывается процесс подготовки тренировочных данных для модели, используемой в модуле распознавания слов.

## 1.2.1 Первоначальная обработка картинок

Прежде чем начать генерировать новые картинки (image augmentation) необходимо определиться с целевыми размерами изображений, которые будут подаваться на вход нейронной сети. Дело в том, что после того как регионы со словами вырезаются из формы, они имеют разные размеры. Это происходит по нескольким причинам. Во-первых, одни и те же слова, написанные разными людьми, могут иметь довольно разную крупность и стиль написания, что косвенным образом влияет на размер регионов слов. Во-вторых, целевые слова имеют большую вариацию между собой по количеству символов. К примеру, в наборе целевых слов имеются такие слова как "Украина" и "Северо-Казахстанская". Здесь первое слово имеет 7 символов, а второе - 20 символов. В-третьих, при вырезании слов, разница между результирующими регионами может колебаться на несколько пикселей.

Хотелось бы отметить, что проблем с определением высоты слова нет, так как мы можем легко привести все собранные изображения слов к одной высоте. Основные проблемы заключаются в определении оптимальной ширины изображения. Мы рассмотрели три варианта решения данной проблемы:

1. *Изменение размера без сохранения пропорций слов.* Размер изображений слов выравнивается таким образом, что не сохраняются пропорции символов на картинке. Данный подход является наиболее простым для реализации. Однако это приводит к нежелательным последствиям. Если говорить более конкретно, то в случае со словами «Украина» и «Северо-Казахстанская» одно из слов будет отображаться некорректно, т.е. будут нарушаться пропорции написания слова. Одно из вышеуказанных слов будет отображаться, как и раньше, а второе будет или растянуто, или сжато. Это зависит от того, какую конечную ширину мы предпочтем – большая ширина изображения означает, что будет храниться большее количество пикселей, а меньшее, соответственно, обратное – картинки будут содержать меньшее количество пикселей. Хотим заметить, что чем меньше пикселей содержит изображение, тем меньше модели может понадобиться вычислительных ресурсов для обучения. Однако, с другой стороны, нельзя пренебрегать точностью отображения объекта на изображении, поэтому необходимо соблюсти баланс между корректностью отображения и размером картинки.
2. *Обрезание изображений.* Данный подход заключается в том, что для всего тренировочного набора данных находится точное положение каждого слова, то есть мы вырезаем слова таким образом, чтобы не оставалось отступов по краям. Далее, проводится масштабирование картинок по высоте. Из новых изображений слов находим то, которое имеет наибольшую ширину (в пикселях). Затем, ширина этого изображения берется за основу. Все изображения слов приводятся к данной ширине, т.е. добавляются пиксели справа и слева таким образом, что непосредственно слово находится по центру региона. Данный подход является более предпочтительным по сравнению с предыдущим методом, так как он сохраняет пропорции символов. Однако, он также является более сложным, так как требует определения точного положения слова на картинке. Определение слов на картинки реализуется с помощью использования алгоритмов нахождения контуров.
3. *Расширение картинки.* Принцип действия данного подхода во многом совпадает с предыдущим методом, но также имеет ряд преимуществ. Как и в методе два, сначала находится точное положение слова на картинке. Оно вырезается. К вырезанному слову применяются различные геометрические трансформации (поворот, смещение, изменение размера и т.д.). Полученный результат записывается на новую картинку с целевым предопределенным размером. Преимущество данного подхода заключается в том, что слово не обязательно должно находиться по центру картинки и быть строго выровнено по горизонтали. С нашей точки зрения, этот подход наиболее приближает наши тренировочные данные к реальным полевым данным, которые будут поступать на вход системы.

Следовательно, мы остановили свой выбор на третьем варианте. Средняя высота и ширина картинок (106, 830). Мы будем приводить все картинки к размеру (64, 512). С нашей точки зрения, данные параметры являются оптимальными с точки зрения сохранения деталей и размера изображений. Мы мотивируем это тем, что в индустрии при работе с печатным текстом обычно используют высоту картинки, не превышающую 32 пикселей. Так как мы имеем дело с рукописным текстом, который отличается от печатной большей случайности, то дополнительные 32 пикселя нам позволят сохранить дополнительные детали почерка.

## 1.2.2 Генерация синтетических данных

Деформация слов в целом*.*

Для генерации новых слов мы использовали геометрические трансформации и добавление шумов (Рисунок 9). В качестве моделей шумов мы использовали аддитивные модели шума: paper-salt model, Gaussian noise model.

В качестве геометрических трансформаций были применены следующие трансформации:

* Эластичная трансформация
* Масштаб
* Перемещение
* Вращение
* Сдвиг влево
* Сдвиг вправо

Так как ячейка в таблице имеет фиксированный размер и почерк у участников эксперимента разный, в большинстве случаев слова расположены слева и имеют пустоту справа от 10% до 70%. Для нормализации входных данных были разработаны следующие методы:

1. Очистка от шумов;
2. Поиск крайних границ текста в картинке;
3. Вырезание картинки по найденным границам;
4. Создание пустой картинки фиксированного размера (512x64)
5. Случайное расположение текста в фиксированном изображении;
6. Добавление шумов случайной величины.

Методы с 4 по 6 позиции были выполнены по 10 раз для каждой входной картины, что дает на выходе 10 раз больше картинок чем было.

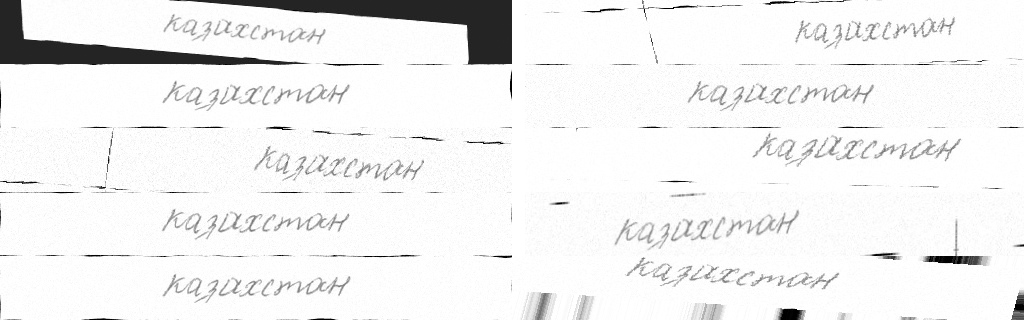


Рисунок 9 – Примеры синтезированных образцов слова

Таким образом все картинки с текстом были предварительно обработаны и увеличены в количестве в 10 раз. Такие данные обеспечат более правильное обучение и исключат проблему местоположения текста как признак и информацию для обучения модели. А шумы сделают модель более гибким для будущих входных изображений.

Генерация слов из символов.

Дополнительно к предыдущему подходу, мы разработали алгоритм по генерации целевых слов из набора рукописных символов. Если говорить более конкретно, то новый метод составляет слова из букв, написанных разными почерками. Символы были собраны с помощью специальных форм. Пример одной из форм показан на рисунке 10.

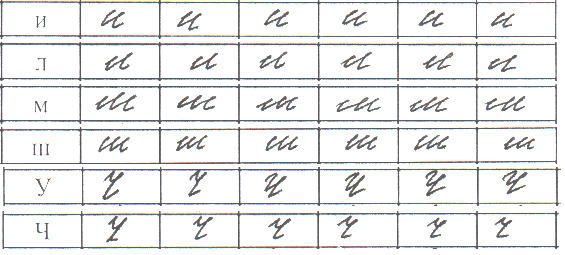


Рисунок 10 – Пример образцов символов в форме

Данный подход генерации новых образцов слов позволяет вносить вариацию в слова на символьном уровне. Другими словами, мы можем, контролируя отступы между буквами и их размер, имитировать различные почерки. Принципиальным отличием данного подхода от предыдущего заключается в том, что последний деформирует все слово в целом.

Новый метод генерации целевых слов позволяет нам значительно увеличить количество тренировочных данных, что, в свою очередь, повышает точность обучаемой модели. Примеры сгенерированных слов с использованием данного метода вы можете увидеть на рисунке 11.

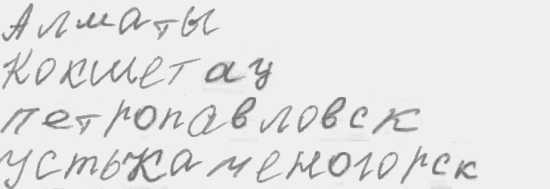


Рисунок 11 – Несколько примеров посимвольно сгенерированных слов

# 2 ПОСТРОЕНИЕ НЕСКОЛЬКИХ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧИ И ПРОВЕДЕНИЕ ОПЫТОВ С КАЖДЫМ РЕШЕНИЕМ

## 2.1 Классификация рукописных слов с использованием Deep Convolutional Neural Networks

В данном эксперименте использовался обычный способ классификации изображений с использованием глубокой сверточной нейронной сети различных моделей. Для правильного распределения данных, были использованы методы предварительной обработки изображения и методы увеличения данных. Для текущего эксперимента, были выбраны десять классов: Казахстан, Белоруссия, Киргизия, Таджикистан, Узбекистан, Астана, Алматы, Актау, Актобе и Атырау.

Эксперимент состоит из трех вариантов:

* простая модель Сверточной нейронной сети [2],
* MobileNet [3],
* MobileNet с небольшими настройками параметров.

*Эксперимент 1*

В данном эксперименте использовалась простая модель Сверточной нейронной сети, которая состоит из двух сверточных слоев и слоя softmax [4] для классификации (Рисунок 12). Такого рода модели часто используются для простых задач, как распознавание символа, классификация животных и так далее.

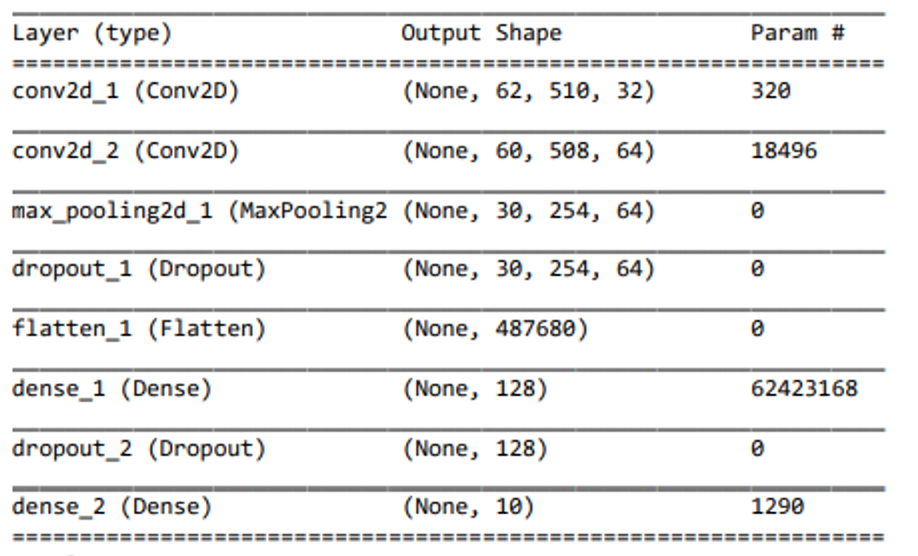


Рисунок 12 – Слои модели, использованные для первого эксперимента

Входными данными для модели были изображения размером 512х64х1. 10% всего датасета были выделены для оценки модели и не участвовали в процессе обучения. Выходными данными являются 10 значений – вероятности распознавания относительное каждого класса. Процесс обучения состояла из 10 итерации и показала следующие результаты (Рисунок 13):

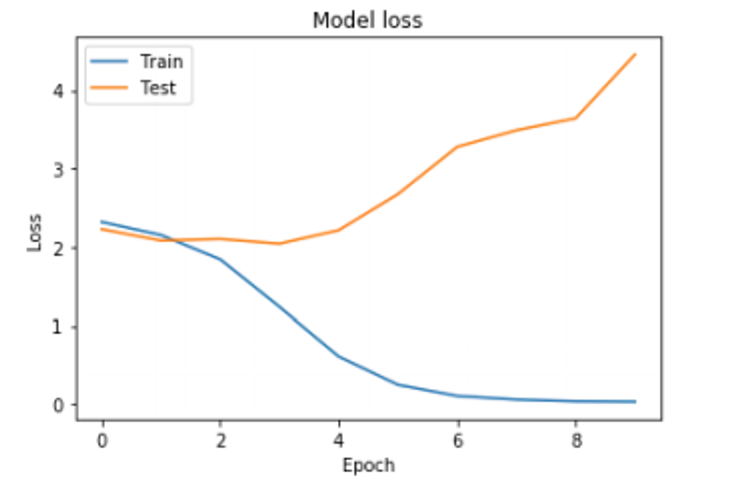
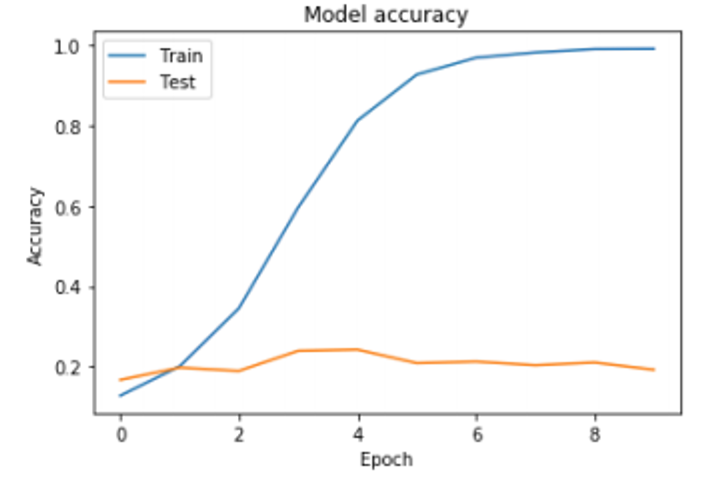


Рисунок 13 – Результаты первого эксперимента: слева точность модели, справа ошибка модели

Как видно из рисунка 13, модель после первой итерации уходит в состояние переобучения, где на данных обучения результаты быстро улучшаются и на тестовых данных наоборот ухудшаются. Это значит, что данная модель не справляется с текущим набором данных и не подходит для решения этой задачи. В связи с этим было решено произвести аналогичный эксперимент с более сложными и глубокими слоями нейронной сети.

*Эксперимент 2 и 3*

Во втором и третьем экспериментах была использована модель, которая называется MobileNet [3]. На рисунке 14 приведена архитектура модели, которая состоит из 30 слоев.

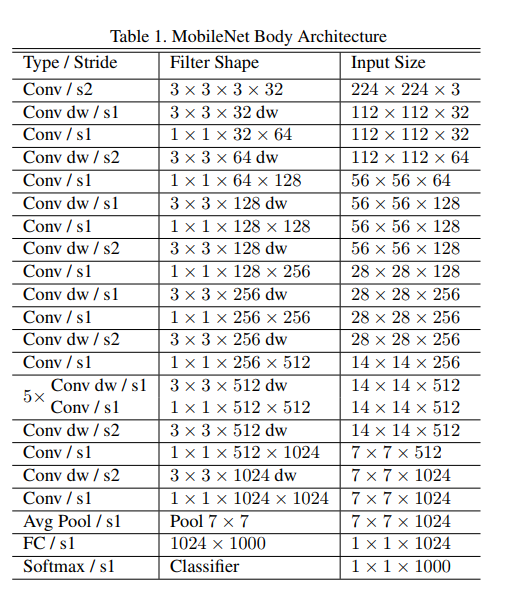


Рисунок 14 – Архитектура MobileNet

Данная модель показала хорошие результаты в задаче ImageNet, где решалась задача классификации 1000 объектов. В текущем эксперименте использовалась модель MobileNet уже с обученными сверточными слоями для задач ImageNet. Были заменены слои, которые отвечают за классификацию из 1000 классов в 10 и переобучены заново с использованием подготовленных ранее рукописных данных. *Эксперимент 2*, как и предыдущий состоял из 10 итераций, и для обучения использовался метод оптимизации Adadelta [4] с начальным шагом обучения learning rate (lr) = 1.0.

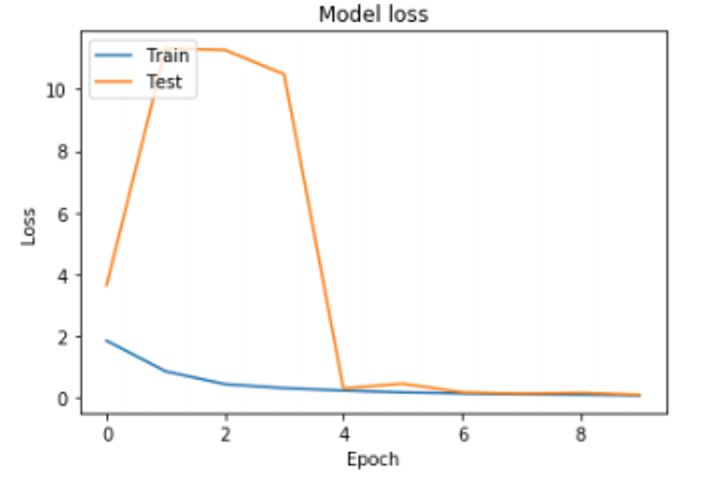
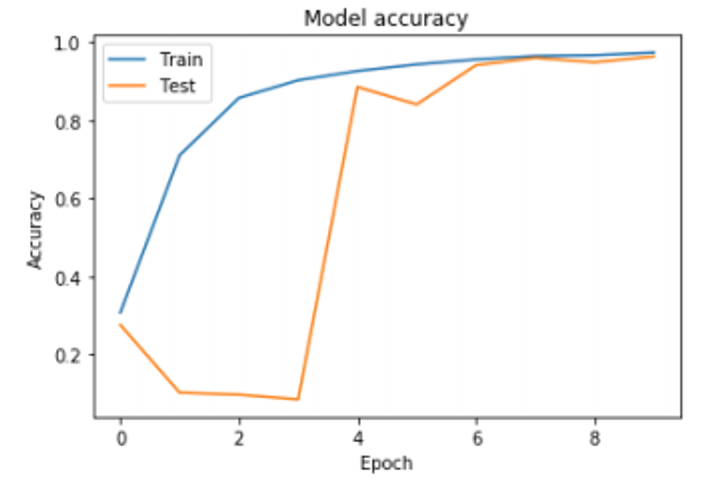


Рисунок 15 – Результаты второго эксперимента с использованием MobileNet (lr=1.0): слева точность модели, справа ошибка модели

На рисунке 15 показаны результаты после 10 итерации обучения и как это видно, модель только после 3-ей итерации начала обучатся правильно. Такое поведение обучения обычно говорит о том, что для метода оптимизации Adadelta начальное значение шага обучения (lr=1.0) очень велик и нужно его уменьшить. Поэтому *Эксперимент 3*проводился в точности как *Эксперимент 2*, только с одним отличием – lr=0.01. Результаты *Эксперимента 3* показаны на рисунке 16.

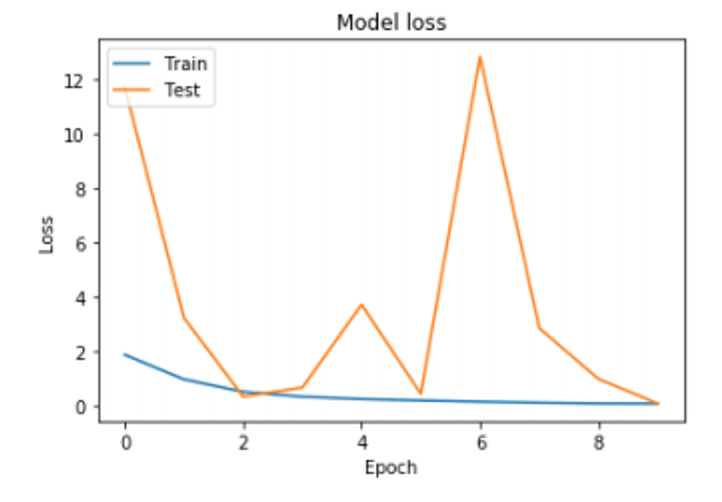
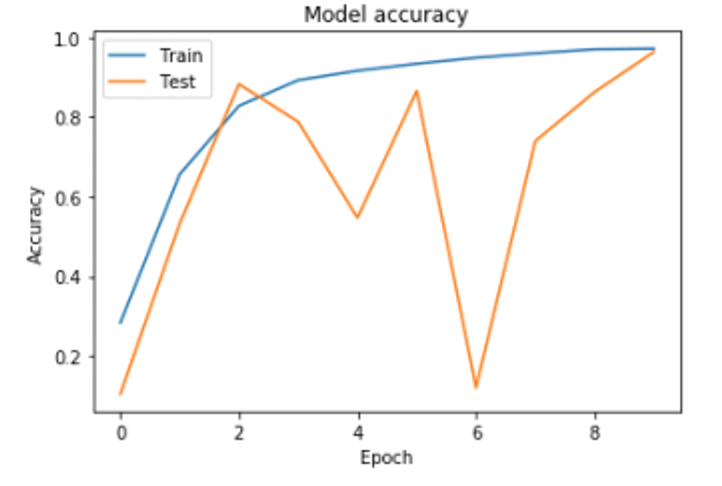


Рисунок 16 – Результаты третьего эксперимента с использованием MobileNet (lr=0.01): слева точность модели, справа ошибка модели

Из рисунка 16 можно увидеть, что начальные итерации стали более правильными. Однако, в шестой итерации модель показывает большой резонанс относительно всего графика. Причины такого поведения все еще изучается, одно из вероятных причин – это малое количество данных. Для проверки данной гипотезы мы запланировали еще больше увеличить данные методами Аффинной трансформации, растягивания изображения и других искажений, и повторить *Эксперимент 3*.

## 2.2 Классификация рукописных слов с использованием модели SimpleHTR

Следующим экспериментом в наших исследованиях была система распознавания рукописного текста SimpleHTR, разработанная Харальдом Шейдл [6]. Предлагаемая система использует Искусственную нейронную сеть (ANN), где несколько слоев Сверточной нейронной сети (CNN) обучаются извлекать соответствующие признаки из входного изображения. Затем эти слои выводят 1D или 2D карту объектов (или последовательность), которая передается слоям Рекуррентной нейронной сети (RNN). RNN распространяет информацию через последовательность. После этого выходные данные RNN отображаются на матрицу, которая содержит оценку для каждого символа на элемент последовательности. Для получения окончательного текста к выходу RNN применяется алгоритм декодирования. Декодирование матрицы выполняется с помощью операции временной классификации соединения (Connectionist Temporal Classiﬁcation – CTC) (Рисунок 17). Для улучшения точности распознования декодирование может использовать также языковую модель.

При обучении с использованием временной классификации соединения (CTC), выход RNN является матрицей, содержащей вероятности символа для каждого временного шага. Алгоритм декодирования CTC преобразует эти символьные вероятности в окончательный текст. Затем для увеличения точности используется алгоритм, которая делает поиск слов в словаре. Однако время прохождения поиска слов зависит от размера словаря и не может декодировать произвольные строки символов, такие как числа.

Операции

CNN: входное изображение подается в слои CNN. Эти слои обучены извлекать соответствующие признаки из входного изображения. Каждый слой состоит из трех операций. Сначала выполняется операция свертки, которая применяет к входу ядро ​​фильтра размером 5×5 в первых двух слоях и 3×3 в последних трех слоях. Затем применяется нелинейная функция RELU. В конце, pooling слой суммирует области изображения и выводит уменьшенную версию ввода. Хотя высота изображения уменьшается на 2 в каждом слое, карты признаков (каналы) добавляются, так что выходная карта признаков (или последовательность) имеет размер 32×256.

RNN: последовательность признаков содержит 256 признаков за временной шаг, RNN распространяет соответствующую информацию через эту последовательность. Используется популярная реализация RNN с длинной кратковременной памятью (LSTM), поскольку она способна распространять информацию на большие расстояния и обеспечивает более надежные характеристики обучения, чем обычная RNN. Выходная последовательность RNN отображается в матрицу размером 32×80. Набор данных IAM состоит из 79 различных символов, далее для операции CTC требуется еще один дополнительный символ (пустая метка CTC), поэтому для каждого из 32 временных шагов имеется 80 записей.

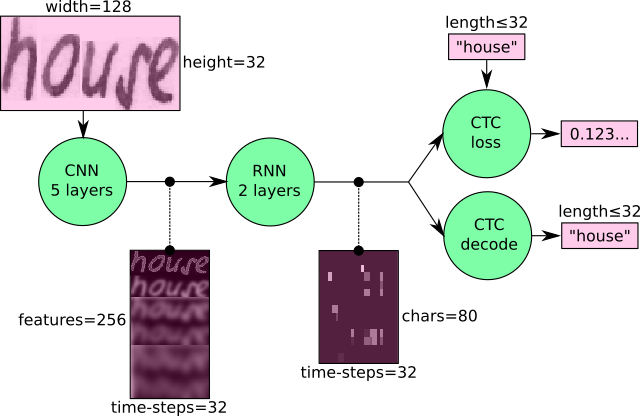


Рисунок 17 – Модель SimpleHTR, где зеленый – операции, а розовый – потоки данных

CTC: во время обучения нейронной сети CTC получает выходную матрицу RNN и желаемый текст, и он вычисляет значение потерь. Получая вывод, CTC получает только матрицу и декодирует ее в окончательный текст. Длина основного текста и распознанного текста может быть не более 32 символов.

Данные

Входные данные: это серое изображение размером 128×32. Обычно изображения из набора данных не имеют точно такого размера, поэтому исходный размер меняется (без искажений), пока он не будет иметь ширину 128 или высоту 32. Затем изображение копируется в белое целевое изображение размером 128×32. Этот процесс показан на рисунке 18. Затем, значения серого нормализуются, что упрощает задачу для нейронной сети.

Реализация SimpleHTR состоит из 4 модулей:

1. *SamplePreprocessor.py*: готовит изображения из набора данных IAM для NN.

2. *DataLoader.py*: считывает образцы, помещает их в пакеты и предоставляет интерфейс итератора для просмотра данных.

3. *Model.py*: создает модель, как описано выше, загружает и сохраняет модели, управляет сеансами Tensorflow и ​​предоставляет интерфейс для обучения и вывода.

4. *main.py*: объединяет все ранее упомянутые модули.

Запуск программы:

*> python main.py*

Аргументы командной строки:

--train: обучение модели,

--validate: валидация модели,

--beamsearch: декодирование методом поиска луча ванили вместо декодирования с наилучшим путем.

--wordbeamsearch: декодирование методом поиска по лучу слов (выводятся только слова, содержащиеся в словаре) вместо декодирования с наилучшим путем.

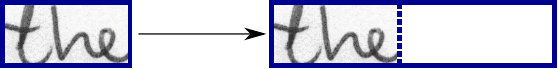


Рисунок 18 – Слева: изображение из набора данных произвольного размера. Масштабируется по размеру целевого изображения размером 128×32, пустая часть целевого изображения заполняется белым цветом

В наших исследованиях мы обучили модель SimpleHTR на собранных нами данных, где имеются 42 наименований стран и городов с различными образцами почерка. Эти данные были увеличены в 10 раз используя методы, описанные в разделе 1.2. Мы провели 2 эксперимента: с выравниванием курсивного написания слов и без выравнивания.

Для того чтобы запустить обучение модели на своих данных были проделаны следующие работы:

1. Подготовлен файл аннотации входных изображений words.txt. Файл содержит следующие строки для каждого изображения:

формат: *0/04010 ok 154 0 0 512 64 NP Казахстан*

где,

*0/04010* -> путь и наименование файла с изображением

*Ok* -> результат сегментации слова:

ok: слово сегментировано правильно,

er: сегментация слова возможна плохая

*154* -> уровень серого для бинаризации строки, содержащей это слово

*0 0 512 64* -> ограничивающий прямоугольник вокруг слова в формате x, y, w, h

*NP* -> грамматическая принадлежность данного слова

*Казахстан* -> транскрипция для этого слова

1. Модифицирован файл *DataLoader.py* для работы с файлом words.txt в кодировке UTF-8 и пути считывания входных изображений.
2. Входные данные были распределены следующим образом: 90% на обучение и 10% на валидацию обученной модели. Из набора данных для обучения 95% используются непосредственно для обучения, и 5% для проверки.
3. После обучения были получены следующие значения на данных валидации:

* с выравниванием курсива:

bestpath: Character error rate: 19.134470%. Word accuracy: 52.555556%

beamsearch: Character error rate: 18.994172%. Word accuracy: 53.333333%

wordbeamsearch: Character error rate: 16.382474%. Word accuracy: 73.555556%.

* без выравнивания:

bestpath: Character error rate: 17.979711%. Word accuracy: 57.111111%

beamsearch: Character error rate: 17.731491%. Word accuracy: 58.333333%

wordbeamsearch: Character error rate: 15.788906%. Word accuracy: 75.111111%

На рисунке 19 продемонстрировано изображение с названием области, которую подали на вход, а на рисунке 20 можем увидеть распознанное слово “Южно-Казахстанская” с вероятностью 86%.

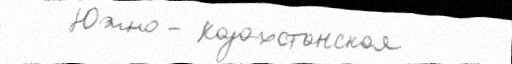


Рисунок 19 – Пример изображения

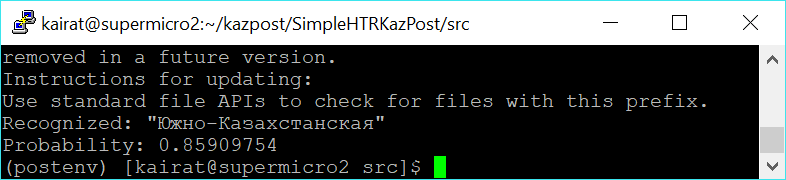


Рисунок 20 – Результат распознавания текста на рисунке 19

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2016. – P.779-788

2 Kim, Yoon. Convolutional neural networks for sentence classification. // arXiv preprint arXiv:1408.5882 (2014).

3 Howard, Andrew G., et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. // arXiv preprint arXiv:1704.04861 (2017).

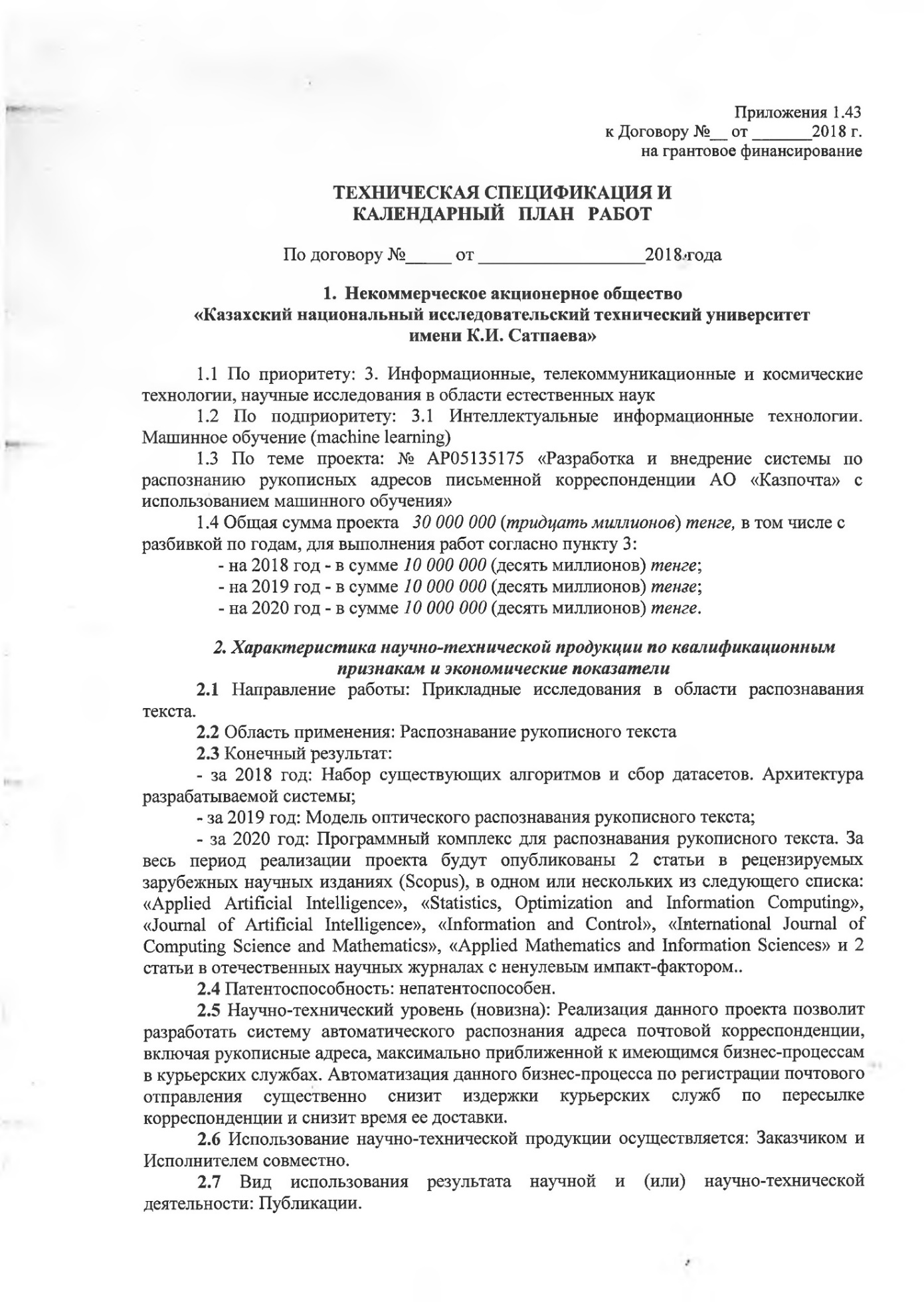
4 Memisevic, Roland, et al. Gated softmax classification. //Advances in neural information processing systems. 2010.

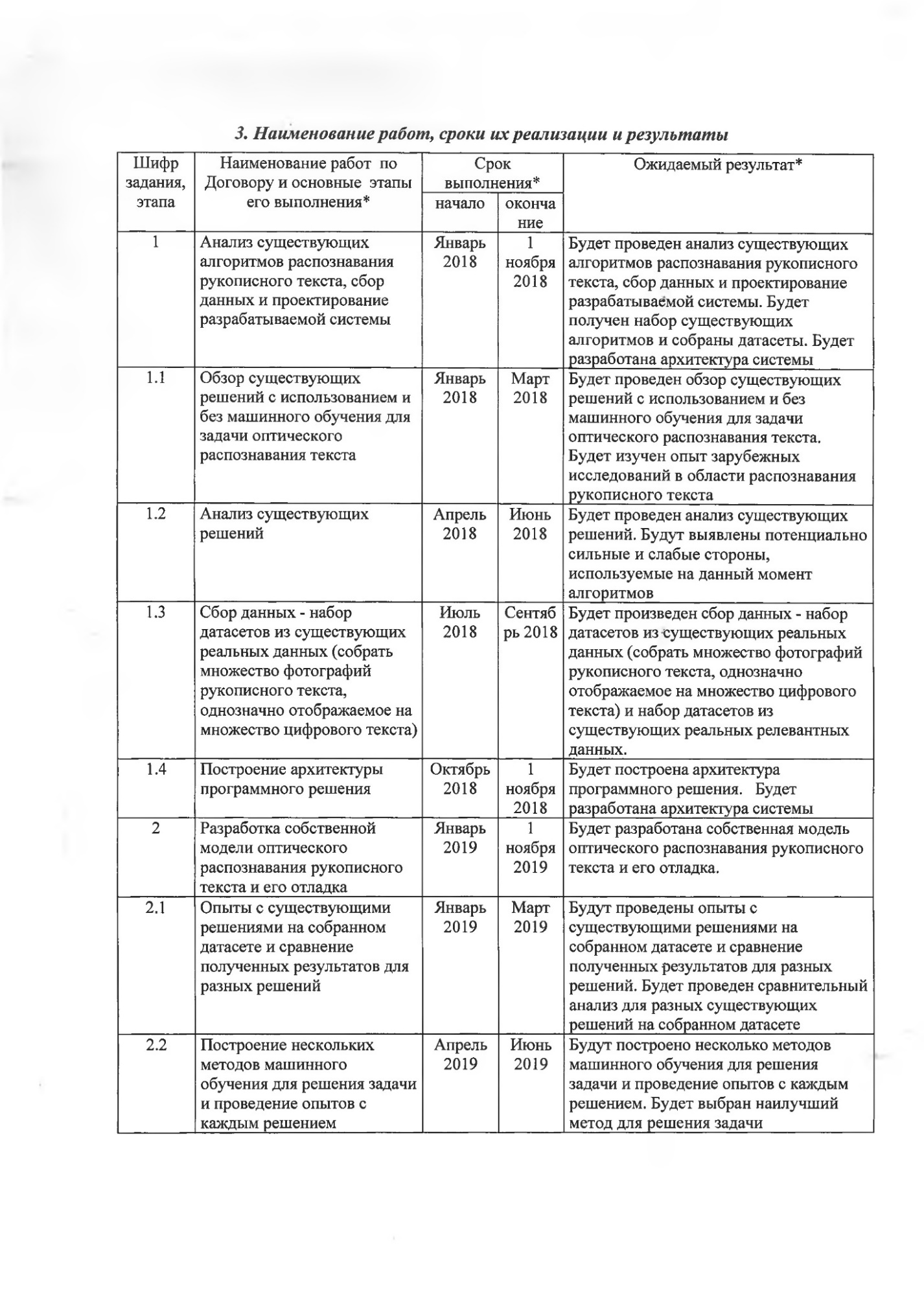
5 Zeiler, Matthew D. ADADELTA: an adaptive learning rate method. // arXiv preprint arXiv:1212.5701 (2012).

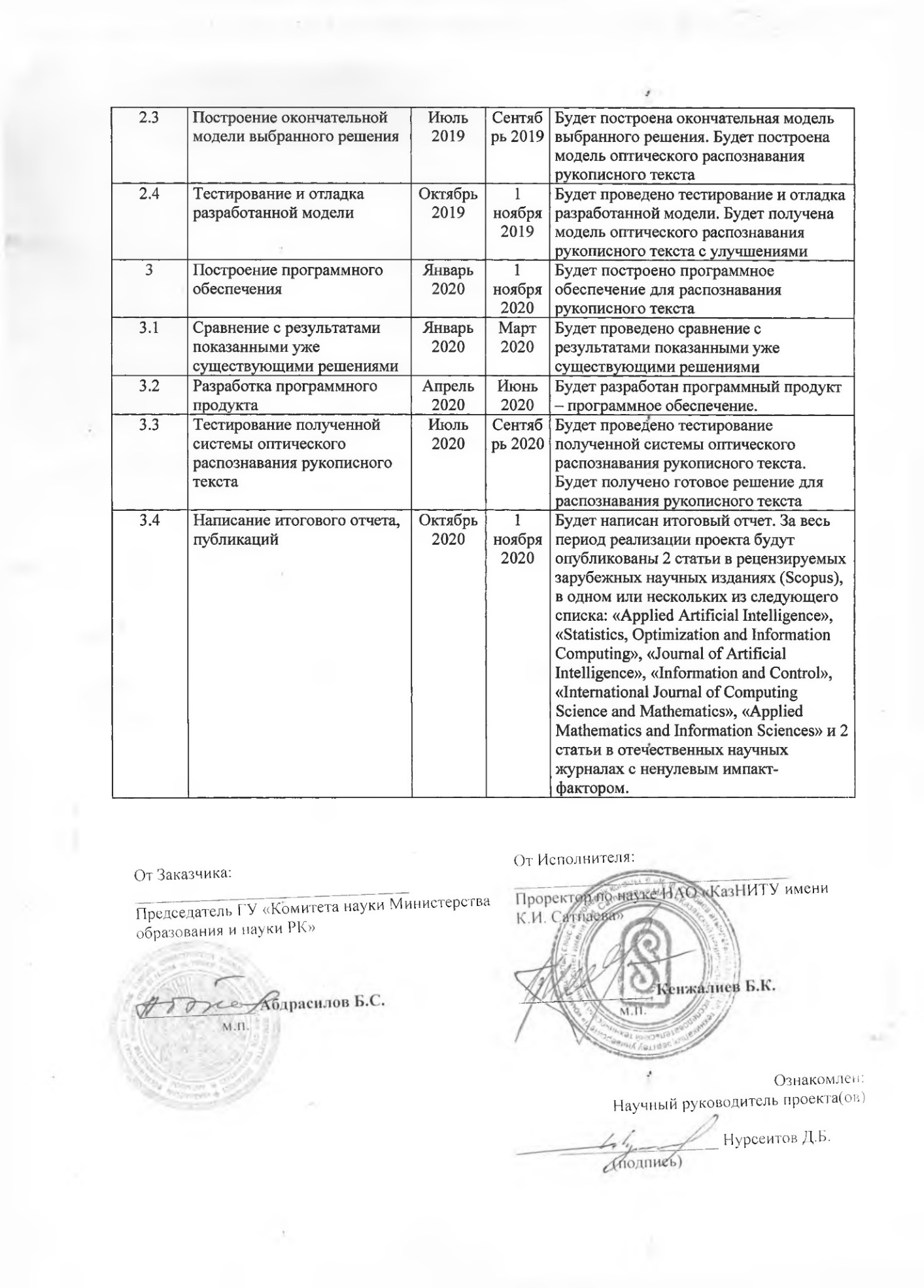
6 Scheidl, Harald. Handwritten text recognition in historical documents. // Diplomarbeit, Technische Universität Wien, 2018 (URN: urn:nbn:at:at-ubtuw:1-115908)

# **ПРИЛОЖЕНИЕ А**

Календарный план

****

****

****

# **ПРИЛОЖЕНИЕ Б**

Список опубликованных работ

1. Нурсеитов Д.Б., Курманходжаев Д.Б., Убайдуллаев Н.М. Модели распознавания рукописных адресов на основе глубокого обучения для сортировки письменной корреспонденции / Материалы XV Международной научно-практической конференции «Эффективные инструменты современных наук – 2019». – Чехия, Прага. – 2019. – С. 40-47