МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАБОТА ПРОВЕРЕНА
Рецензент,
Технический директор
ООО "Компания СГ-групп"
_____ Д.В. Гапонюк
“___”_____ 2018 г.

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой,
д.ф.-м.н., профессор
_____ Л.Б. Соколинский
“___”_____ 2018 г.

РАЗРАБОТКА СИСТЕМЫ ОБРАБОТКИ МАТЕМАТИЧЕСКИХ РУКОПИСНЫХ ФОРМУЛ С ПРИМЕНЕНИЕМ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2018.115-006.ВКР

Научный руководитель
к.ф.-м.н., доцент кафедры СП,
_____ В. Голодов

Автор работы,
студент группы КЭ-401
_____ Д.И. Валеев

Ученый секретарь
(нормоконтролер)
_____ О.Н. Иванова
“___”_____ 2018 г.

Челябинск-2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Искусственные нейронные сети (ИНС) (Artificial neural networks, ANN) — упрощенные модели биологических нейронных сетей [26].

Сверточные нейронные сети (СНС, CNN) очень похожи на обычные нейронные сети: они также построены на основе нейронов, которые обладают изменяющимся весом и смещениями. Каждый нейрон получает некоторые входные данные, выполняет скалярное произведение информации и в отдельных ситуациях сопровождает это нелинейностью. Как и в случае с обычными нейронными сетями, вся CNN выражает одну дифференцируемую функцию взноса (эффективный взнос): с одной стороны, это необработанные пиксели изображения, с другой — вывод класса или группы вероятных классов, характеризующих картинку [29].

Классификация — один из разделов машинного обучения, посвященный решению следующей задачи. Имеется множество объектов (ситуаций), разделенных некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется обучающей выборкой. Классовая принадлежность остальных объектов не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества [25].

Задача сегментации может быть сформулирована как разбиение исходного изображения на множество непересекающихся связных областей, ассоциируемых с объектами наблюдаемой сцены или их частями в соответствии с некоторыми выбранными критериями [30].

LaTeX — наиболее популярный набор макрорасширений системы компьютерного набора TeX [24].

АКТУАЛЬНОСТЬ ТЕМЫ ИССЛЕДОВАНИЯ

Задача распознавания печатного и рукописного текста приобрела актуальность в связи с широким распространением компьютерной техники.

Существует множество решений задачи распознавания текста, вплоть до онлайн-сервисов [22]. Уровень распознавания существующих инструментов позволяет автоматизировать обработку машинописного текста и свести к минимуму необходимость постобработки человеком. Например, такими возможностями обладает программа *ABBYY FineReader* [27] и программа *InftyReader* [4]. Обе эти программы практически точно распознают печатные математические формулы. Однако в распознавании рукописных математических формул этот уровень еще достигнут.

С 2011 по 2016 год проводилось ежегодное соревнование по распознаванию математических формул — CROHME (*Competition on Recognition of Online Handwritten Mathematical Expressions*). Последнее подобное соревнование было проведено в 2016 году. Победу одержала команда «WIRIS» [11]. Для классификации отдельных символов ими были использованы технологии *нейронных сетей*.

Идея распознавания символов с помощью искусственных нейронных сетей рассматривалась Яном Лекуном и другими исследователями в работе «Handwritten character recognition using neural networks architecture» [8]. Задача распознавания символов является задачей классификации. Для решения этой задачи в основном используются *сверточные нейронные сети*.

Задача распознавания машинописного текста/формул и рукописного текста практически решена, однако распознавание рукописных математических выражений, по-прежнему, является актуальной задачей на сегодняшний день. Наиболее перспективным подходом к решению данной задачи является применение нейросетевых технологий**.**

ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ

Целью данной работы является разработка системы обработки математических рукописных формул, перевода рукописных формул в цифровые форматы LaTeX, MathML. Система должна быть реализована в виде веб-сервиса.

Для осуществления поставленной цели необходимо решить следующие задачи.

1. Провести обзор существующих аналогов и научной литературы по распознаванию рукописного текста и математических формул.

2. Подготовить обучающую и тестовую выборку рукописных символов.

3. Разработать нейронную сеть для распознавания рукописных математических символов.

4. Провести тестирование нейронной сети.

5. Реализовать систему обработки рукописных математических формул в виде веб-сервиса и провести ее тестирование.

СТРУКТУРА И ОБЪЕМ РАБОТЫ

Работа состоит из введения, 5 разделов, заключения, библиографии и трех приложений. Объем работы составляет 43 страницы, объем библиографии — 30 источников, объем приложений — 5 страниц.

В первой главе производится анализ предметной области и обзор существующих аналогов в данной области. Также здесь приведены существующие решения для создания нейронных сетей.

Во второй главе приведены описания алгоритмов, используемых в программной системе.

В третьей главе содержатся требования к классам нейронной сети, функциональные требования к программной системе, а также варианты использования этой системы.

Четвертая глава содержит детали реализации системы распознавания рукописных математических формул.

В пятой главе приведены результаты тестирования нейронной сети и системы распознавания рукописных математических формул.

В заключении приводятся основные результаты работы и рассматриваются направления дальнейших исследований.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Распознавание рукописных математических формул

Распознавание математических выражений является более сложной задачей распознавания, чем распознавание символов, т.к. кроме распознавания непосредственно математического символа необходимо также распознать структуру математического выражения.

В научной литературе разделяют два вида распознавания рукописных символов: **онлайн и оффлайн.**

Онлайн распознавание подразумевает непосредственное рисование формулы на компьютере в режиме реального времени. При этом ведется сбор дополнительной информации в виде времени и координат каждого штриха.

Оффлайн распознавание подразумевает отсутствие информации о ходе написания символа, дано только изображение содержащее формулу, на котором формулу требуется найти и распознать, т.е. выделить символы на фона, разбить на строки и т.д.

В данной работе будет рассматриваться задача оффлайн распознавания.

В исследовании [13] приводится список источников данных, которые можно использовать для обучения нейронной сети: база данных MNIST [23] с 60 тыс. рукописных цифр, база данных Detexify с более чем 200 тыс. греческих букв и математических символов. Для распознавания отдельно взятого символа использовалась сверточная нейронная сеть на базе LeNet.

В статье [20] рассматривается проблема схожести большого количества символов в LaTeX и о том, как такие проблемы решать. Для распознавания символов были использованы технологии нейронных сетей.

В статье [8] описывается разработка архитектуры нейронной сети для распознавания рукописных цифр. Процент ошибок этой нейронной сети составляет 1 %.

1.2. Обзор аналогичных проектов и существующих решений

Рассмотрим существующие решения в области распознавания математических формул.

Detexify [2]

Данный веб-сервис позволяет рисовать символ онлайн, а затем распознает его и предоставляет данные наиболее вероятных символов. Пример работы сервиса представлен на рисунке 1.

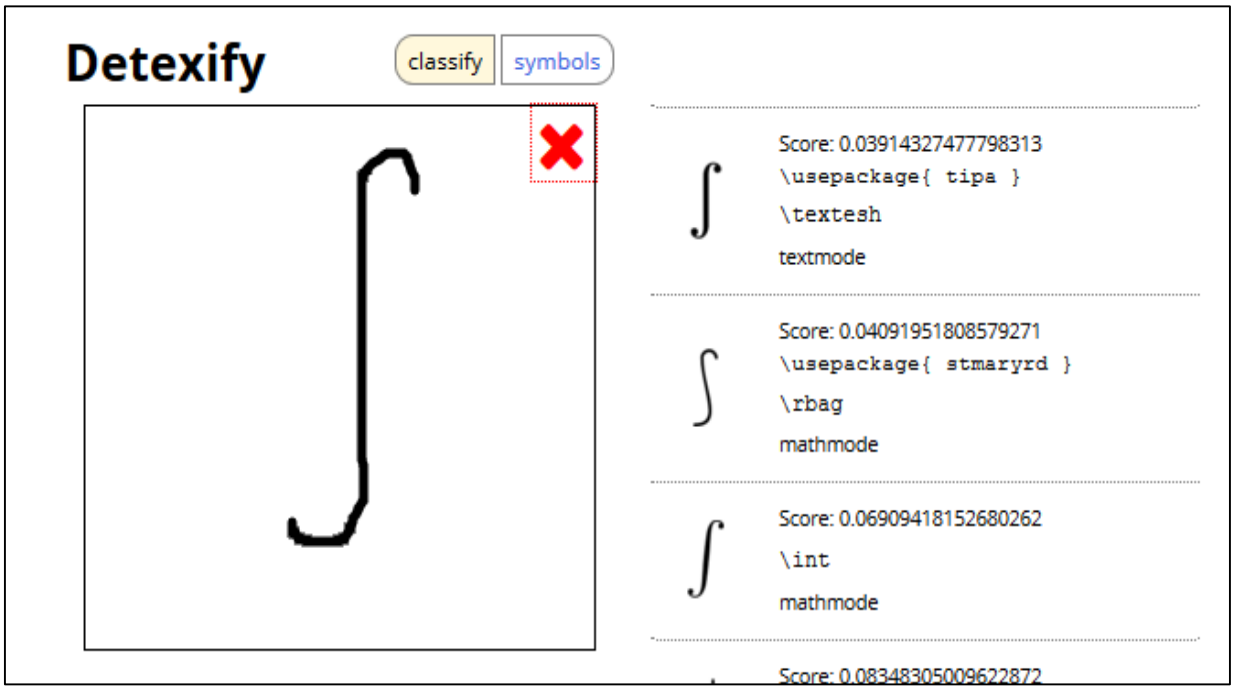

Рис. 1. Пример работы веб-сервиса Detexify

Стоит отметить, что данный сервис отлично справляется со своей задачей. Также его исходный код полностью открыт и выложен на GitHub [6].

Недостатком данного сервиса является то, что производится распознавание только одного символа. Также значительным минусом является невозможность загрузить свое изображение.

Shapecatcher [17]

Этот сервис очень схож по своему функционалу с предыдущим. Главным различием является то, что здесь происходит распознавание Unicode-символов.

Как и в предыдущем сервисе, возможно распознавание лишь одного символа. И так же только нарисованного онлайн. Пример работы сервиса представлен на рисунке 2.

Рис. 2. Веб-сервис Shapecatcher в действии

Google Docs [3]

В Документах Google также представлена подобная возможность распознавания нарисованного онлайн символа. Выглядит и работает так же, как и в двух предыдущих сервисах.

Этот инструмент очень удобен для поиска и вставки в документ необходимого вам специального символа. Пример работы представлен на рисунке 3.

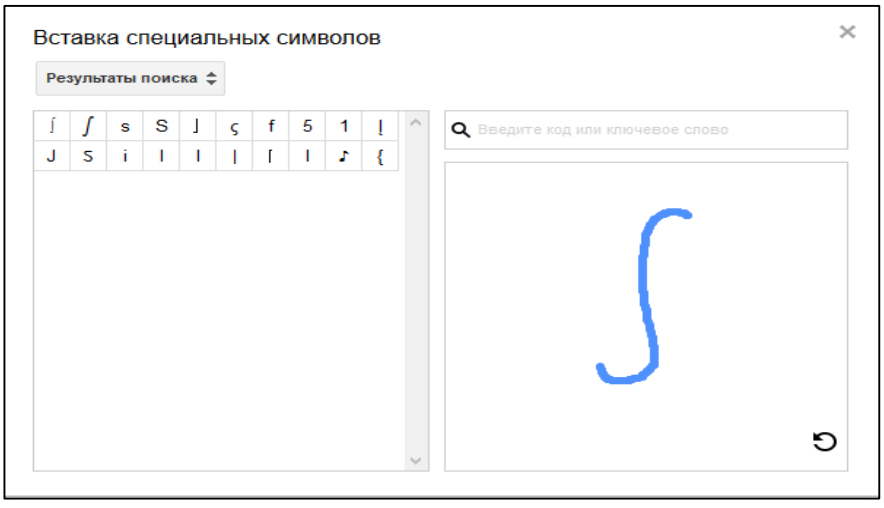

Рис. 3. Распознавание специальных символов в Google Docs

Mathematical Expression Recognition [9]

Данный сервис, в отличие от всех предыдущих, предоставляет возможность распознавания нескольких символов. На выходе система выдает LaTeX-код. Пример работы сервиса вы можете увидеть на рисунке 4.

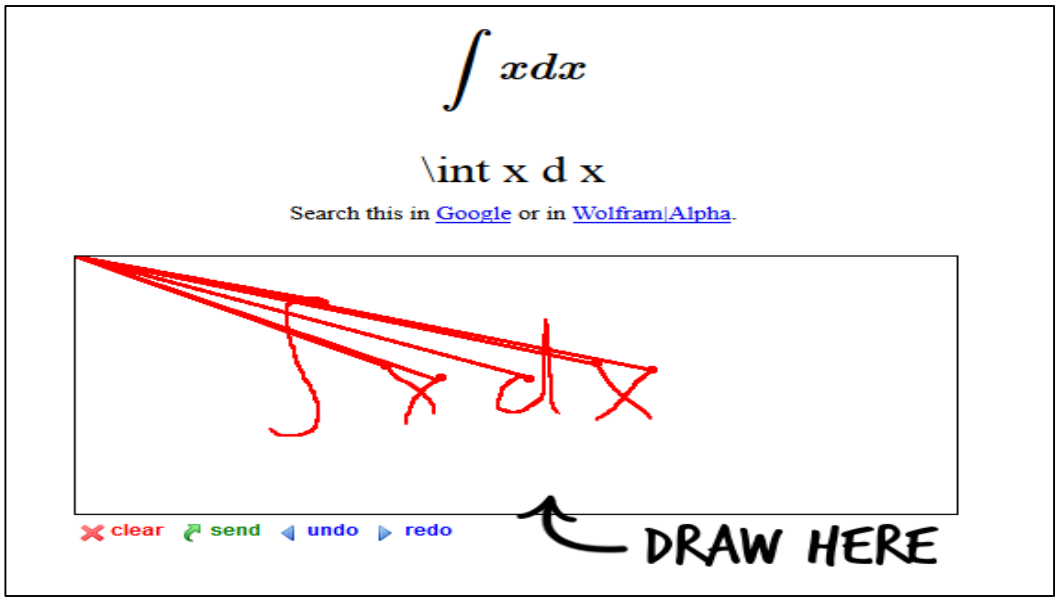

Рис. 4. Пример работы MER

Photomath [14]

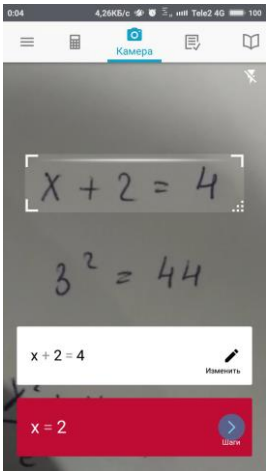

Рис 5. Пример работы приложения Photomath

Photomath — это приложение для Android, позволяющее автоматически решать простые математические примеры, просто засняв их на камеру. Система автоматически распознает формулу на фотографии, а также про-

водит ее вычисление. Умеет решать лишь простые примеры, в одну строчку, без интегралов и т.д. Пример представлен на рисунке 5.

Сравнение аналогов

В таблице табл. 1 представлен сравнительный анализ всех аналогов, рассмотренных выше.

Табл. 1. Сравнение аналогов

Аналог / Критерий	Detexify	Shapecatcher	Google Docs	MER	Photomath
Онлайн	★	★	★	★	-
Оффлайн	-	-	-	-	★
LaTeX	★	-	-	★	-
Unicode	-	★	★	-	-
Распознавание нескольких символов	-	-	-	★	★
Распознавание дробных выражений	-	-	-	★	★
Распознавание выражений с интегралами	-	-	-	★	★
Пошаговое решение	-	-	-	-	★

Большая часть аналогов представляет собой онлайн распознавание. Единственным представителем оффлайн распознавания в этом списке является мобильное приложение Photomath, которое обладает широким функционалом, но не выдает формулу в виде LaTeX-кода.

1.3. Обзор готовых решений для создания нейронных сетей

Первая сверточная нейронная сеть для распознавания символов была представлена в 1998 году французским исследователем Яном Лекуном (Yann LeCun) [7]. Называется она *LeNet*. Структура данной сети представлена на рисунке ниже.

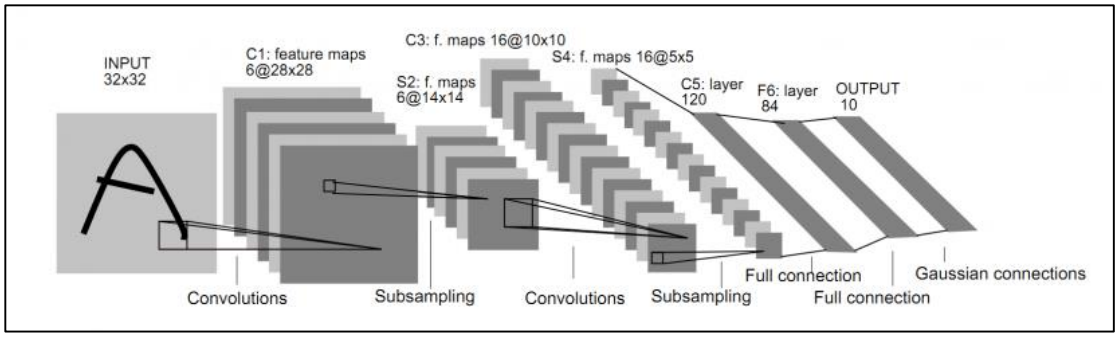

Рис. 6. Структура нейронной сети LeNet

Для языка программирования Python существует большое количество готовых библиотек для работы с нейронными сетями. Использование этих библиотек призвано в значительной степени упростить задачу разработки нейронных сетей. Далее в этом разделе будут рассмотрены наиболее популярные из них.

TensorFlow [18]

Очень популярная библиотека, разрабатываемая Google. Документация для данной библиотеки хорошо проработана. Также очень много статей в сети Интернет и примеров нейронных сетей для данной библиотеки. Для представления нейронной сети здесь используются графы, а для хранения данных многомерные массивы – тензоры.

Theano [19]

Другая не менее популярная библиотека для Python. Библиотека получила свое название в честь имени жены древнегреческого философа и математика Пифагора — Феано (или Теано). Так же, как и других библиотеках, имеется интеграция с библиотекой NumPy [12]. Присутствует возможность проведения вычислений как на CPU, так и на GPU.

Keras [5]

Библиотека, а если быть точнее, фреймворк для создания нейронных сетей. Keras, является надстройкой над TensorFlow и Theano и может использовать любую из этих двух библиотек для проведения вычислений. Из всех библиотек эта наиболее удобна и проста для понимания. Новые слои здесь добавляются лишь с использованием одной функции. Хорошая документация и множество готовых примеров – все это является большим преимуществом для Keras.

По итогам обзора библиотек было решено использовать Keras, как наиболее понятную и простую для новичков. Его функционала нам будет достаточно для построения необходимой нам нейронной сети.

1.4. Соревнование по распознаванию математических формул

Соревнование CROHME проводилось с 2011 по 2016 год. С каждым годом результаты и сложность распознавания увеличивалась. Вместе с этим росло и количество данных, используемых для обучения и тестирования разработанных систем. Так, например, в 2011 году для обучения использовалось 921 математическое выражение и 348 для тестирования [10]. Тогда как, в 2016 это количество данных достигло 8836 и 1147 для обучения и для тестирования соответственно.

Вместе с тем, как росло количество данных, увеличивалось и сложность, и качество распознавания. Максимальная точность распознавания основной задачи (рукописных математических формул) в 2011 году составляла 22,41 %, такой результат показала команда из организаторов конкурса, которая в самом соревновании не участвовала. В 2016 году лучший результат по данной задаче уже был 67,65 %, этот результат был продемонстрирован командой «MyScript».

Соревнование состоит из нескольких задач. В 2016 году их было четыре, некоторые из которых имели подзадачи. Они были определены следующим образом.

1. Распознавание формул. Распознавание рукописных математических формул (основная задача).

2. Распознавание символов. Классификация отдельных математических символов.

3. Распознавание структуры. Распознавание структуры математической формулы.

4. Матричное распознавание. Распознавание выражений, содержащих матрицы из рукописных штрихов. Данная задача являлась экспериментальной.

Исходные данные (математические выражения) хранятся в файлах с форматом inkML. Это простые текстовые файлы, написанные в XML-формате. InkML-файл состоит из 3 частей: координаты штрихов, истина на уровне символов, истина на уровне математического выражения.

Некоторые команды использовали собственные, *приватные,* данные для обучения. В 2016 году команде MyScript, таким образом удалось достигнуть точности 67,65 %. Для этого были использованы около 30 000 собственных математических формул. В то же время, команда WIRIS, используя лишь предоставленные организаторами данные, достигла результата 49,61 %.

Выводы по первому разделу

Для задачи распознавания рукописных математических формул уже существует ряд подходов и их реализации. Однако большинство из них имеет существенные недостатки. Например, не позволяют загрузить собственное изображение и не предоставляют удобный графический интерфейс для работы. Также, большая часть этих инструментов не имеет удобного вывода результата в различные форматы (LaTeX и т.п.). Необходимо будет учесть эти недостатки при разработке собственного приложения.

Развитие и популярность нейросетевых технологий является дополнительным стимулом к тому, чтобы проводить исследование в этой обла-

сти. Также имеется множество существующих библиотек для создания и обучения нейронных сетей, которые облегчают эту задачу. Стоит принять во внимание и тот факт, что на соревновании CROHME активно применяются нейросетевые технологии. Один из лучших результатов показала команда «MyScript» с точностью 67,65 %. Тем самым задача распознавания рукописных математических формул с применением нейросетевых технологий является актуальной.

2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1. Сверточные нейронные сети

Сверточные нейронные сети, как архитектура искусственных нейронных сетей, были предложены Яном Лекуном (Yan LeCun) в 1988 году. Эта архитектура направлена на эффективное распознавание изображений. Пример сверточной нейронной сети представлен на рисунке 7.

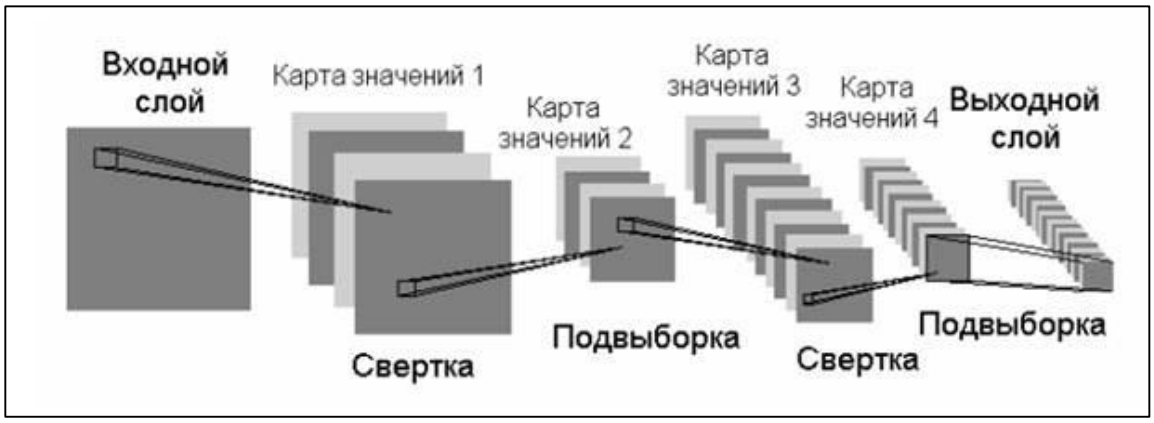

Рис. 7. Пример сверточной нейронной сети

Основной идеей является чередование слоев свертки и слоев подвыборки. За счет того, что учитывается двумерность изображения, сверточные нейронные сети показывают себя лучше на распознавании изображений, чем другие виды искусственных нейронных сетей.

Алгоритм работы слоя свертки заключается в следующем, каждый фрагмент исходного изображения умножается на матрицу (ядро) свертки, полученные результаты суммируются и записываются в исходное положение, но уже на другом изображении. Этот процесс изображен на рисунке 8.

Также, помимо слоев свертки, используются слои подвыборки. Такой слой позволяет уменьшить размерность изображения. Считается, что наличие признака, важнее его положения. Суть метода заключается в том, что группа пикселей (например, 2 на 2) сжимается до одного. Обычно для этого используется функция максимума. Пример работы слоя подвыборки представлен на рисунке 9.

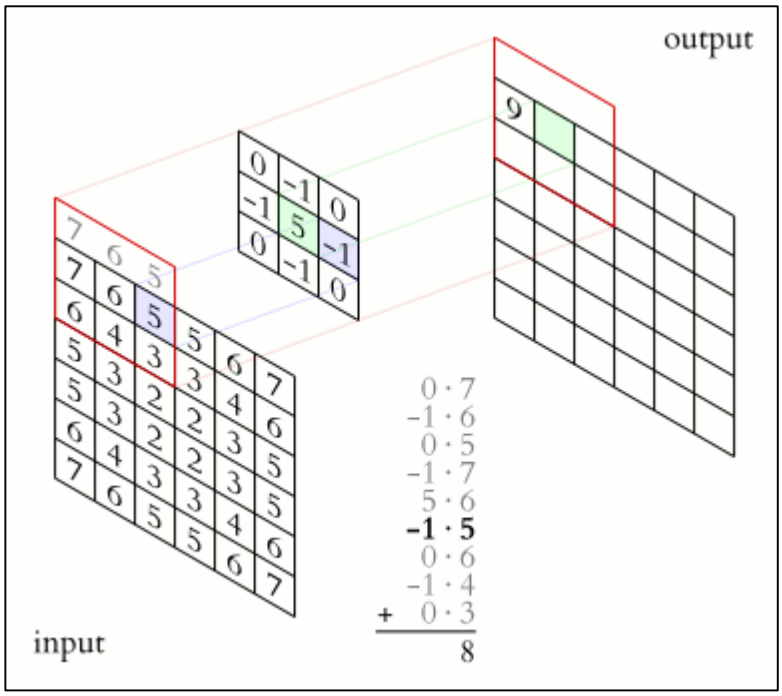

Рис. 8. Пример свертки изображения

После прохождения нескольких слоев свертки и подвыборки, количество каналов увеличивается, а размерности изображений уменьшаются. На этом этапе происходит переход к полносвязным слоям. В результате, последний слой имеет размерность, соответствующую числу классов.

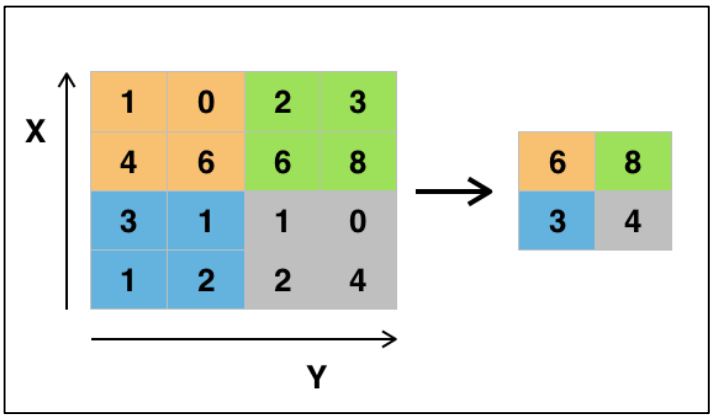

Рис. 9. Пример работы слоя подвыборки с функцией максимума

2.2. Алгоритм BFS (breadth-first search)

Алгоритм BFS или поиск в ширину является методом обхода графа. Его основная суть заключается в том, что происходит последовательный обход уровней графа, который начинается с вершины-источника.

Пошаговое выполнение алгоритма.

1. Пометить вершину, с которой начинается поиск, и поместить в пустую очередь.
2. Извлечь из очереди вершину и пометить ее.
3. Добавить в очередь все соседние вершины, которые еще не помечены и не добавлены в очередь.
4. Если очередь пуста, значит все вершины просмотрены.
5. Иначе возвращаемся к пункту 2.

Выводы по второму разделу

Была рассмотрена теоретическая часть о сверточных нейронных сетях: алгоритм работы, информация о слоях свертки и подвыборки. Также был рассмотрен алгоритм обхода в ширину.

3. ПРОЕКТИРОВАНИЕ

Система распознавания рукописных математических формул будет представлять собой веб-сервис. С помощью этого сервиса у пользователя будет возможность загрузить изображение, распознать на нем формулы и получить готовый результат в формате LaTeX.

Требования к классам нейронной сети

В конечный набор классов должны входить:

- цифры (10 штук);
- буквы латинского алфавита верхнего и нижнего регистров (52 штуки);
- буквы греческого алфавита (48 штуки);
- математические символы: +, -, =.

Итого: 113 классов, каждый класс отвечает за один символ.

Требования к системе распознавания формул

На основании анализа предметной области и описания теоритической части были определены следующие функциональные требования к разрабатываемой системе.

1. Пользователь должен иметь возможность выбрать и загрузить изображение.
2. Пользователь должен иметь возможность получить готовый результат в формате LaTeX.
3. Система должна загружать пользовательское изображение.
4. Система должна распознавать символы на изображении и формировать их в готовую формулу.

3.1. Варианты использования системы распознавания формул

Для проектирования приложения был использован язык графического описания для объектно-ориентированного программирования UML. Бы-

ла построена модель взаимодействия внешнего актера с программной системой в виде диаграммы вариантов использования (use-case diagram).

Полученная диаграмма изображена на рисунке 10.

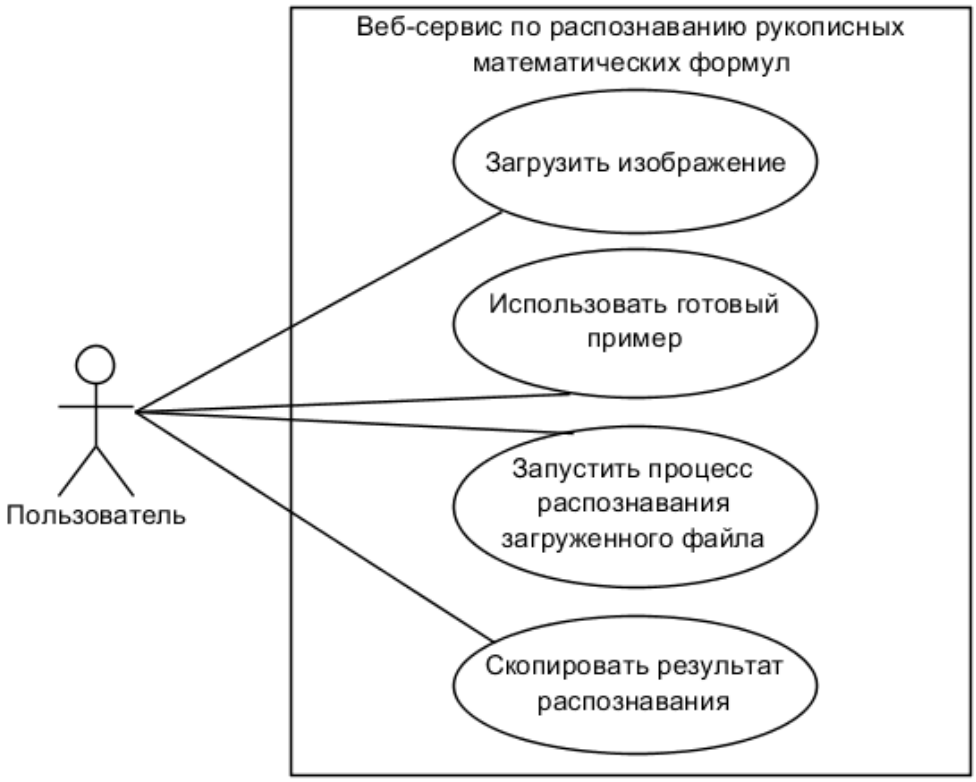

Рис. 10. Диаграмма вариантов использования

На представленной диаграмме изображен 1 актер:

Пользователь — человек, пользующийся системой и использующий определенные функции.

Для каждого актера определены свои варианты использования.

Пользователю доступны следующие функции:

1) загрузить изображение — пользователь выбирает изображение для распознавания и загружает его на сервер;

2) использовать готовый пример — вместо загрузки собственного изображения, пользователь может использовать готовый пример;

3) запустить процесс распознавания загруженного файла — пользователь нажимает на кнопку «Распознать» и начинается процесс распознавания;

4) скопировать результат распознавания — пользователь может скопировать результат распознавания в специальных текстовых полях.

Диаграммы деятельности

На основе требований к системе была разработана диаграмма деятельности, представленная на рисунке 11.

Эта диаграмма показывает, как происходит процесс взаимодействия пользователя с системой.

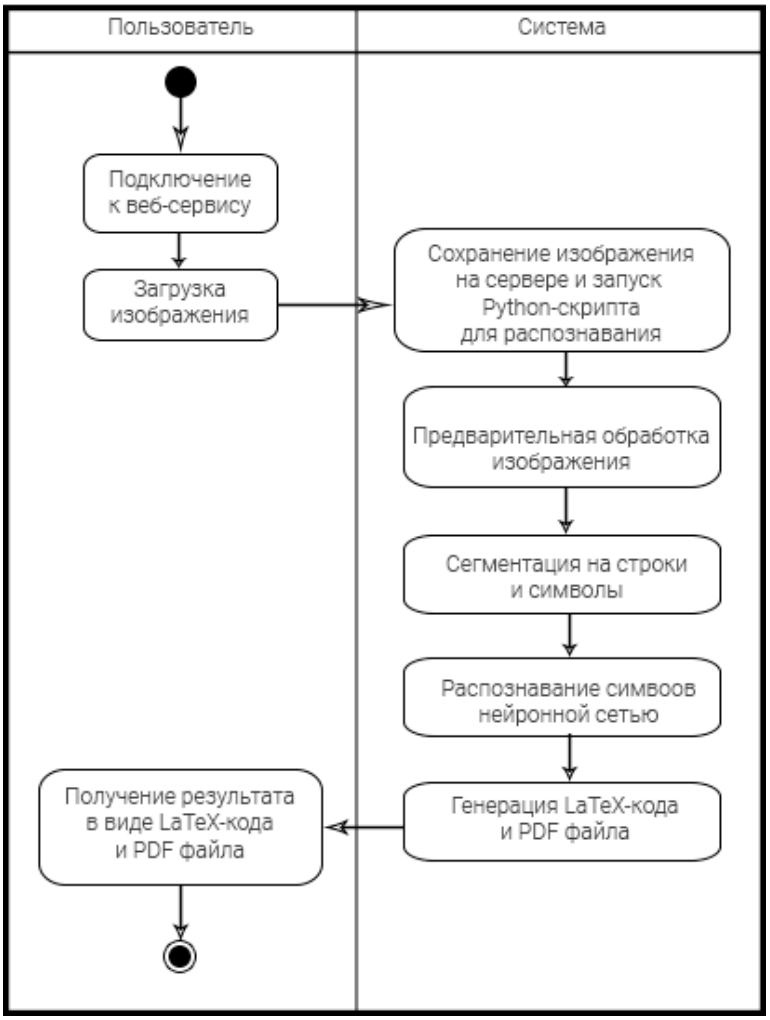

Рис. 11. Диаграмма деятельности

3.2. Проектирование графического интерфейса пользователя

Интерфейс веб-сервиса должен быть простым и понятным, но при этом должен полностью обеспечивать необходимую функциональность.

Интерфейс веб-сервиса состоит из 3 основных частей: шапка сайта, основная секция и подвал. Основная секция содержит поле для загрузки

изображения пользователем, и результат распознавания. Результат выводится в виде таблицы, в которой содержатся: исходное изображение, результат в виде PDF файла, результат в виде LaTeX-кода.

Макет графического интерфейса пользователя изображен на рисунке 12.

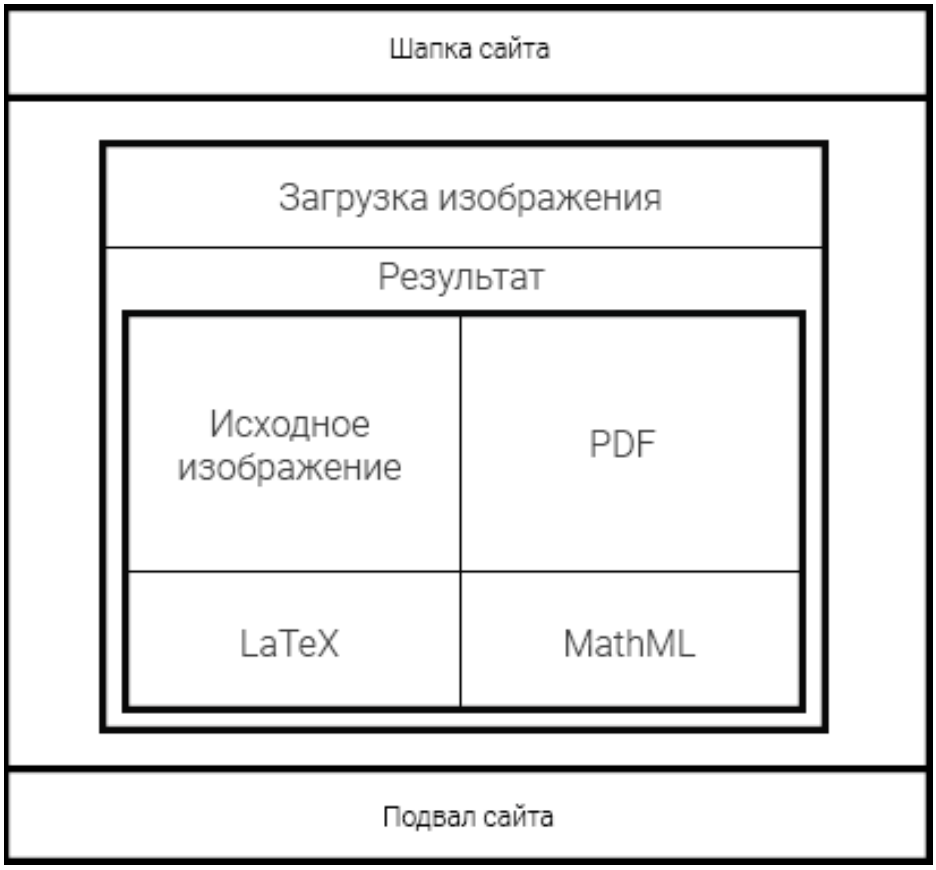

Рис. 12. Макет графического интерфейса

Выводы по третьему разделу

Были определены требования к классам нейронной сети и требования к системе распознавания рукописных математических формул. На основе этих требований была спроектирована диаграмма вариантов использования, диаграмма деятельности, а также разработан макет графического интерфейса для веб-сервиса.

4. РЕАЛИЗАЦИЯ

Средства разработки

Для разработки программной части для распознавания формул был использован высокоуровневый язык программирования Python 3.5. Разработка велась в среде разработки PyCharm Professional 2017.1 [16] и на операционной системе Deepin Linux 15.5 [1].

Для языка программирования Python было использованы стандартные библиотеки 2Python, а также Keras, Tensorflow, Pillow [15], tkinter [21] и т.д.

Для реализации веб составляющей были использованы технологии:

1. HTML5, CSS3 для верстки.
2. Javascript, jQuery 3.x для интерактивности и посылки Ajax-запросов.
3. PHP 7 на стороне сервера для работы сервиса и для перехвата и обработки Ajax-запросов, а также для запуска Python-скрипта для распознавания формул.

4.1. Формирование обучающей выборки

Для обучения распознаванию цифр была использована база данных MNIST. Она содержит 60 000 изображений рукописных цифр.

Для распознавания математических символов и букв греческого алфавита была использована база данных Detexify.

Для распознавания букв английского алфавита была собрана собственная база из более чем 5 000 символов. Для этой цели был разработан специальный веб-сервис, который описан ниже.

В связи с тем, что некоторые буквы в верхнем и нижнем регистре в английском и греческом алфавите очень похожи друг на друга, было решено убрать 12 букв из английского и 15 букв из греческого. Также из числа классов был удален символ «равно», т.к. по сути это два символа «минус», находящихся друг под другом.

Таким образом, для обучения остается 85 классов, среди которых нет похожих символов. Итоговый список символов можно увидеть на рисунке 13.

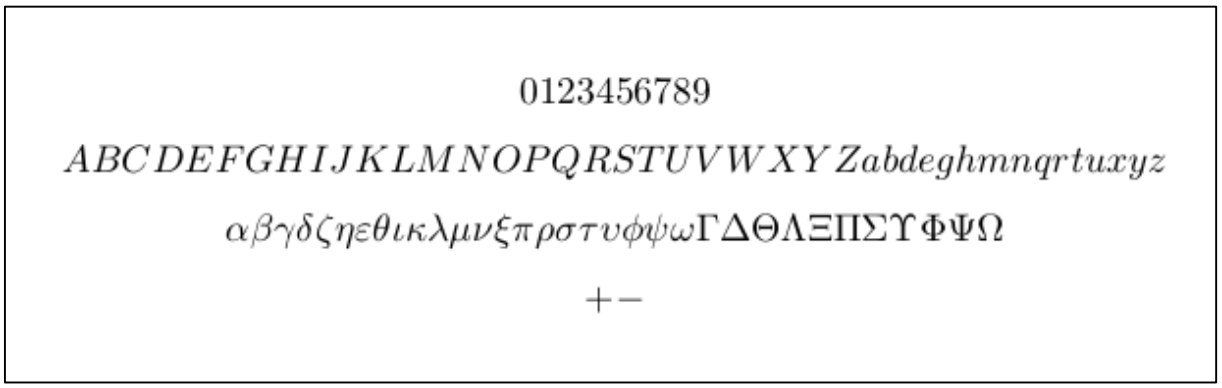

Рис. 13. Визуальное представление классов

Итоговое количество символов для обучения составило 91 562 символа.

Предобработка данных для обучающей выборки

В связи с тем, что данные для обучающей выборки собирались из разных источников, они имеют разный формат. В базе данных MNIST – это изображения, размером 28x28 пикселей. В базе данных Detexify все данные хранятся в виде координат штрихов. Веб-сервис сохраняет символы в виде изображений 400 на 400.

Все эти данные необходимо привести к одному формату. Был выбран формат изображения 28x28 пикселей. Причины следующие:

1) не придется преобразовывать данные MNIST, увеличивать такие изображения нецелесообразно;

2) изображение размером 28x28 пикселей достаточно для корректного отображения любого символа из алфавита;

3) изображение размером 28x28 пикселей достаточно небольшое и обучение нейронной сети будет происходить быстрее.

В следующих двух подразделах будет рассмотрена подготовка данных из этих источников.

Работа с базой данных Detexify

Detexify – онлайн-сервис для распознавания математических символов.

База данных Detexify содержит несколько сотен тысяч математических символов. Доступ к базе данных был предварительно запрошен у владельца, в результате чего был получен SQL-файл (весом 1Гб) со всей базой. В котором содержится более чем 200 тысяч символов.

В качестве системы управления базами данных был выбран PostrgreSQL. Для работы с базой данных была использована библиотека *py-postgresql* для Python.

Символы в базе данных хранятся в виде координат в диапазоне от (0, 0) до (400, 400) пикселей. Всего имеется 3 столбца: *id*, *key* и *strokes*. Столбец *id* – уникальный идентификатор, *key* – классификация символа (например, «latex2e-OT1-_beta»), *strokes* – это координаты точек, по которым рисует символ и дополнительно к ним значение *timestamp* (временной промежуток, используется для определения последовательности рисования точек).

Для нейронной сети были необходимы изображения размером 28x28 пикселей, поэтому было необходимо конвертировать данные из базы в изображения, а затем в текстовые данные.

Для работы с большим количеством символов были разработаны специальные утилиты на языке Python. Одна из них считывала необходимые нам данные из базы данных и превращала в изображения размером 400 пикселей в ширину и в высоту.

Алгоритм преобразования данных из БД в изображение.

1. Координаты считываются из базы и записываются в массив.
2. Массив сортируется по столбцу *timestamp*.
3. С помощью библиотеки *pillow* символ рисуется поточечно, при этом соединяя точки между собой линиями (толщина линии зависит от разброса координат).

4. Полученное изображение размером 400x400 сохраняется в папку с соответствующим именем.

Далее полученные изображения преобразуются в необходимый формат. Они приводятся к размеру 28x28 пикселей.

Веб-сервис для пополнения обучающей выборки

Для обучения нейронной сети была необходима большая обучающая выборка. В связи с этим, было принято решение создать веб-сервис для сбора необходимого количества символов.

Итоговый вариант сервиса представлен на рисунке 14.

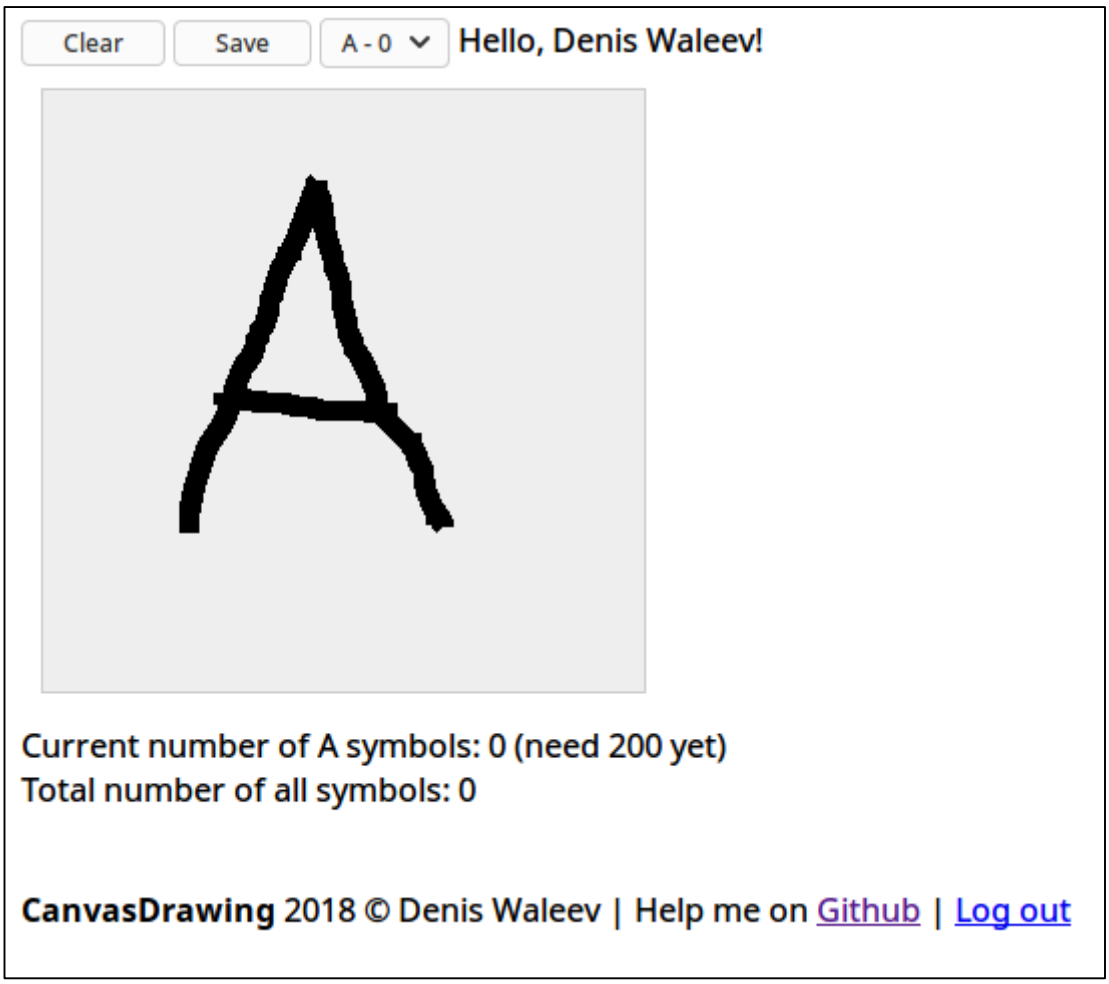

Рис. 14. Скриншот веб-сервиса

При разработке сервиса использовались следующие технологии.

1. Клиентская часть: HTML 5; CSS 3; Javascript (jQuery, AJAX, canvas2D).

2. Серверная часть: PHP 5.3; MySQL.

Принцип работы очень прост. Пользователь выбирает из списка определенный символ, затем рисует его в специальном для этого месте. Чтобы отправить готовый символ на сервер необходимо нажать кнопку «Send image» или клавишу пробел.

Изображение в автоматическом режиме загружается и сохраняется на сервере в специальную директорию.

С помощью данного сервиса производился сбор данных для английского алфавита верхнего и нижнего регистра. Всего было собрано более 5000 изображений, которые потом в автоматическом режиме были преобразованы в необходимый для нейронной сети вид.

Стоит отметить, что в одиночку собрать такое количество данных непросто, поэтому были привлечены люди, которые и нарисовали большую часть собранных изображений.

Для того, чтобы процесс рисования символов был не таким скучным, на сайте была реализована таблица лидеров. Чем больше символов нарисовал пользователь, тем выше в таблице он находился. Такая таблица была дополнительным мотивирующим фактором, для того, чтобы рисовать все больше и больше.

4.2. Топология нейронной сети

Для распознавания рукописных математических формул была разработана модель в виде сверточной нейронной сети, которая представлена на рисунке 15. За основу была взята модель LeNet-5, разработанная Яном Лекуном.

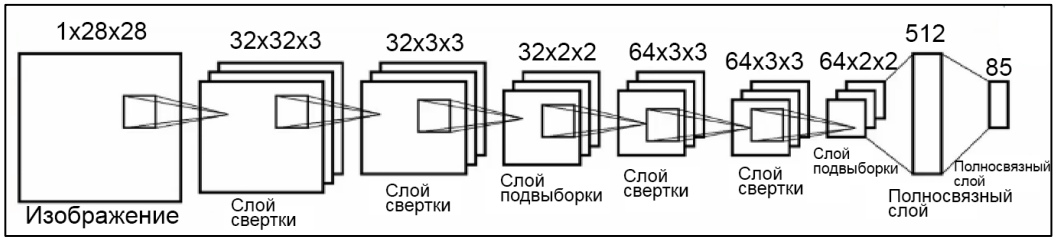

Рис. 15. Структура сверточной нейронной сети

Полученная нейронная сеть состоит из:

- 4 слоев свертки;
- 2 слоев подвыборки;
- 2 полносвязных слоев.

На входной слой подается черно-белое изображение, размером 28x28 пикселей. На выходе – полносвязный слой, размером, соответствующим количеству классов.

4.3. Распознавание формулы

В начале происходит подготовка изображения. В частности, изображение бинаризуется. Затем, происходит поиск связных компонент в изображении. Слишком маленькие по размеру убираются. Далее, каждый символ подготавливается к распознаванию, и подается на распознавание нейронной сети. Результаты распознавания нейронной сети собираются и происходит постобработка символов. На данном этапе, как раз, и собирается формула. Далее распознанная формула преобразуется в нужный формат и выводится пользователю.

В следующих подразделах будет детально освещен каждый этап распознавания формулы.

Подготовка изображения

На данном этапе происходит предварительное улучшение исходного изображения. Оно бинаризуется и из него убирается лишний шум.

Алгоритм сегментирования

Для выделения символов из фона используется алгоритм BFS.

После того, как были выделены все области связности, некоторые из них отбрасываются, в частности, те, которые являются слишком маленькими по размеру (шум).

Затем, для каждого символа определяется его положение, границы, точка центра. Эта информация будет использована в дальнейшем.

Подготовка символа к распознаванию

Для того, чтобы распознать изображения с символами с помощью нейронной сети, их следует подготовить, т.е. преобразовать в нужный формат.

Процесс подготовки символа к распознаванию можно описать так.

1. Символ вырезается по краям из исходного изображения.
2. Изображение преобразуется в квадратное по большему краю.
3. Размеры полученного изображения преобразуются в 20x20.
4. По краям добавляется пустое пространство и размер увеличивается до 28x28 пикселей.

Описанный процесс представлен на рисунке 16.

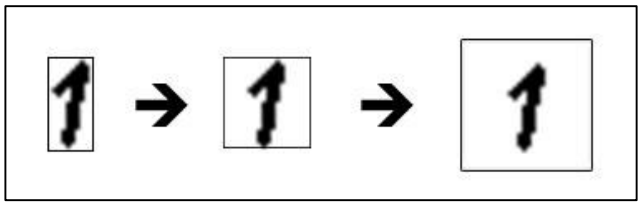

Рис. 16. Преобразование символа

Полученное изображение преобразуется в одномерный массив размером 784 элемента (28x28 пикселей). И нормируется, то есть значения от 0 до 255 преобразуются в значения в интервале от 0 до 1.

Распознавание символов

На данном этапе происходит распознавание каждого отдельно взятого символа с помощью нейронной сети.

Определение положения символа в формуле

После того, как символы были распознаны, проводится их дополнительная обработка.

Символы сортируются слева направо, и далее оценивается их принадлежность к степени или к нижнему индексу. Для этого вычисляется угол между центрами текущего и предыдущего символа.

1. Если угол больше 30 и меньше 60 градусов, то это степень.

2. Если угол больше -60 и меньше -30 градусов, то это нижний индекс.

Визуально это можно увидеть на рисунке 17.

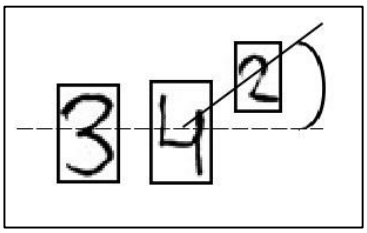

Рис. 17. Определение положения по углу

Далее обрабатываются все символы, распознанные как символ «минус». И если два минуса идут подряд, причем они находятся друг под другом, то два данных символа преобразуются в один – «равно».

Соединение отдельных символов в формулу

После того как были обработаны все символы, необходимо их собрать в одну формулу.

Символы собираются последовательно, слева направо. Для каждого символа определяется его LaTeX-код и положение в формуле.

4.4. Разработка веб-сервиса

Для того, чтобы процесс распознавания формул был максимально удобным был разработан веб-сервис. Он позволяет распознавать формулы онлайн, не устанавливая для этого отдельных программ.

Веб-сервис располагается на облачных серверах AWS (Amazon Web Services). На сервере установлен дистрибутив Ubuntu 16.04 LTS.

Предварительно на сервер были установлены все необходимые библиотеки для языка Python. Также был установлен XAMPP.

Дизайн веб-сервиса

Для веб-сервиса, согласно макету (рис. 12), был нарисован и свёрстан шаблон на HTML5 и CSS3. Для добавления интерактивности была использована javascript-библиотека jQuery.

Результат представлен на рисунке 18.

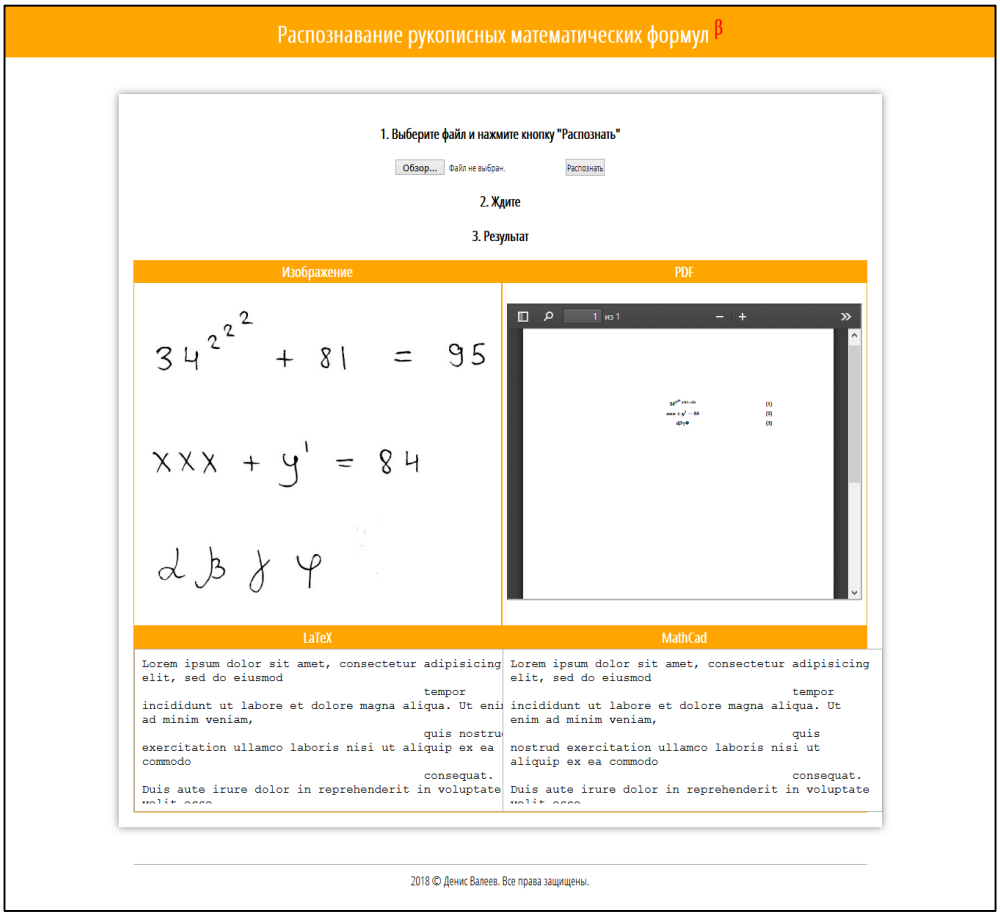

Рис. 18. Шаблон веб-сервиса

Во время того, как пользователь находится в ожидании результата распознавания, он видит gif-изображение имитирующее загрузку (рисунок 19).

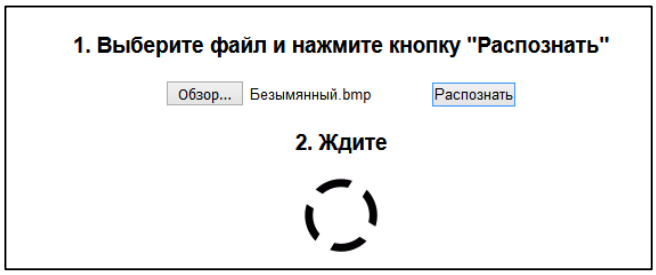

Рис. 19. Имитация процесса распознавания

Настройка сервера AWS

Для начала на сервере были установлены все необходимые библиотеки для Python.

Затем для работы веб-сервиса был установлен XAMPP (кроссплатформенная сборка веб-сервера). Вместе с XAMPP были установлены PHP, MySQL и т.д.

Для удобной работы над проектом был создан репозиторий на bitbucket.org [28]. Таким образом все локальные изменения заливались сначала в репозиторий, затем оттуда на облако.

Принцип работы

Кратко о схеме работы веб-сервиса. Пользователь загружает свое изображение. Изображение при помощи Ajax передается на сервер, где его обрабатывает PHP-скрипт. Изображение загружается на сервер, а ссылка на него передается в Python-скрипт для распознавания, запуск которого также происходит в PHP-скрипте.

Далее результат распознавания обрабатывается и передается обратно пользователю.

Для того, чтобы каждый мог протестировать веб-сервис без загрузки собственного изображения, была добавлена возможность использовать тестовые изображения.

Скриншоты, демонстрирующие работу веб-сервиса представлены в приложении 1.

Домен recognizemath.tk

Для веб-сервиса был подключен бесплатный домен верхнего уровня в сфере «.tk». Для этого домен сначала был делегирован на DNS сервера Яндекса, и затем в редакторе DNS были прописаны IP сервера AWS.

Выводы по четвертому разделу

На основе разработанной архитектуры был реализован веб-сервис для распознавания рукописных математических формул. Разработанный веб-сервис соответствует предложенным требованиям.

5. ТЕСТИРОВАНИЕ

Тестирование нейронной сети

Сверточная нейронная сеть была обучена на 20 эпохах. При обучении нейронная сеть показала значение точности 0,95, а функция потерь 0,16.

В качестве функции подсчета потерь использовалась категорийная кросс-энтропия (categorical cross-entropy), а для оптимизации – стохастический градиентный спуск (Stochastic Gradient Descent).

Функциональное тестирование

Было выполнено функциональное тестирование согласно функциональным требованиям, представленным в разделе 3.2. Результаты тестирования приведены ниже.

Тест № 1. Цель: проверка запуска.

Ожидаемый результат: веб-сервис должен загружаться и выводить шаблон в браузере.

Вывод: ожидаемый и полученный результат совпадают, цель достигнута.

Тест № 2. Цель: проверить загрузку изображения и запуск процесса распознавания.

Ожидаемый результат: веб-сервис должен загрузить выбранное изображение и запустить процесс распознавания.

Полученные результаты изображены на рисунках в приложении.

Вывод: ожидаемый и полученный результат совпадают, цель достигнута.

Тест № 3. Цель: проверить процесс распознавания символов на изображении.

Ожидаемый результат: при нажатии на кнопку «Распознать», приложение должно начать процесс распознавания символов на изображении и по окончании процесса, выдать результат пользователю.

Вывод: ожидаемый и полученный результат совпадают, цель достигнута.

Тест № 4. Цель: проверить автоматическое выделение текста в текстовых полях с LaTeX и MathML кодом.

Ожидаемый результат: при клике на текстовое поле, текст автоматически выделяется.

Вывод: ожидаемый и полученный результат совпадают, цель достигнута.

Тестирование на реальных примерах

Для тестирования были подготовлены несколько сканов рукописных документов. Все изображения и полученные результаты представлены в приложении 2.

Вывод по пятому разделу

Было проведено функциональное тестирование в соответствии с требованиями. Сверточная нейронная сеть также была протестирована на основе специального набора символов. Приведено тестирование веб-сервиса на специальном наборе изображений.

ЗАКЛЮЧЕНИЕ

В рамках дипломного проекта был разработан веб-сервис для распознавания рукописных математических формул.

В ходе разработки были решены следующие задачи.

1. Был проведен обзор существующих аналогов и научной литературы по распознаванию рукописного текста и математических формул.
2. Была подготовлена обучающая и тестовая выборка рукописных символов.
3. Разработана нейронная сеть для распознавания рукописных математических символов.
4. Проведено тестирование нейронной сети.
5. Реализована система обработки рукописных математических формул в виде веб-сервиса и проведено ее тестирование.

Работа апробирована на XIII Уральской выставке научно-технического творчества молодежи «Евразийские ворота России». Получила высокую оценку и диплом первой степени за лучшую творческую работу по информационным технологиям с вручением памятной медали выставки. Диплом представлен в приложении 3.

Дальнейшее направление работы

Планируется продолжать разработку и улучшение веб-сервиса, в частности.

1. Улучшить процент распознавания символов нейронной сетью путем увеличения обучающей выборки.
2. Увеличить количество классов в обучающей выборке.
3. Увеличить скорость распознавания.
4. Повысить удобство использования веб-сервиса.

ЛИТЕРАТУРА

1. Deepin. Официальный сайт. [Электронный ресурс] URL: https://www.deepin.org/ru/ (дата обращения: 12.01.2018).

2. Detexify LaTeX handwritten symbol recognition. [Электронный ресурс] URL: http://detexify.kirelabs.org/ (дата обращения: 15.02.2018).

3. Google Документы. [Электронный ресурс] URL: http://docs.google.com/ (дата обращения: 15.02.2018).

4. InftyReader. Официальный сайт. [Электронный ресурс] URL: http://www.inftyreader.org/ (дата обращения: 22.02.2018).

5. Keras. Официальная документация. [Электронный ресурс] URL: https://keras.io/ (дата обращения: 13.01.2018).

6. kirel/detexify: Latex Symbol Classifier Web Frontend. [Электронный ресурс] URL: https://github.com/kirel/detexify (дата обращения: 18.02.2018).

7. LeCun Y. LeNet-5, convolutional neural networks. [Электронный ресурс] URL: http://yann.lecun.com/exdb/lenet/ (дата обращения: 19.02.2018).

8. Matan O. Handwritten Character Recognition Using Neural Network Architectures. / O. Matan, R. Kiang, C. Stenard, et al. // Proceedings of the 4th US Postal Service Advanced Technology Conference. – 1990. – Vol. 9. – P. 1003-1011.

9. Mathematical Expression Recognition. [Электронный ресурс] URL: http://cat.prhlt.upv.es/mer/ (дата обращения: 20.02.2018).

10. Mouchere H. CROHME2011 : Competition on Recognition of Online Handwritten Mathematical Expressions. / H. Mouchere, C. Viard-Gaudin, D. H. Kim, et al. // 2011 International Conference on Document Analysis and Recognition. – 2011. – Vol. 4. – No. 1. – P. 1497–1500.

11. Mouchere H. ICFHR 2016 CROHME : Competition on Recognition of Online Handwritten Mathematical Expressions. / H. Mouchere, C. Viard-Gaudin, R. Zanibbi, et al. // 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). – 2016. – Vol. 5. – No. 1. – P. 607–612.

12. NumPy. Официальный сайт. [Электронный ресурс] URL: http://www.numpy.org/ (дата обращения: 18.01.2018).

13. Offline Mathematical Expression Recognition. [Электронный ресурс] URL: http://vision.soic.indiana.edu/b657/sp2016/ (дата обращения: 18.02.2018).

14. Photomath. Приложение на Google Play. [Электронный ресурс] URL: https://play.google.com/store/apps/details?id=com.microblink.photomath (дата обращения: 28.02.2018).

15. Pillow. Официальная документация. [Электронный ресурс] URL: http://pillow.readthedocs.io/ (дата обращения: 11.02.2018).

16. PyCharm. Официальный сайт. [Электронный ресурс] URL: https://www.jetbrains.com/pycharm/ (дата обращения: 18.09.2017).

17. Shapecatcher: Unicode Character Recognition. [Электронный ресурс] URL: http://shapecatcher.com/ (дата обращения: 18.02.2018).

18. TensorFlow. Официальный сайт. [Электронный ресурс] URL: https://www.tensorflow.org/ (дата обращения: 15.01.2018).

19. Theano. Официальная документация. [Электронный ресурс] URL: http://deeplearning.net/software/theano/ (дата обращения: 11.02.2018).

20. Thoma M. On-line Recognition of Handwritten Mathematical Symbols. // Bachelor Thesis, 2015. – 102 p.

21. Tkinter. Официальная документация. [Электронный ресурс] URL: https://docs.python.org/3/library/tkinter.html (дата обращения: 21.01.2018).

22. Write Math. [Электронный ресурс] URL: http://write-math.com/ (дата обращения: 20.02.2018).

23. База данных MNIST. [Электронный ресурс] URL: http://yann.lecun.com/exdb/mnist/ (дата обращения: 14.02.2018).

24. Документация LaTeX. [Электронный ресурс] URL: http://www.latex-project.org/help/documentation/ (дата обращения: 20.02.2018).

25. Классификация. [Электронный ресурс] URL: http://www.machinelearning.ru/wiki/index.php?title=Классификация (дата обращения: 20.02.2018).

26. Нейронные сети. Изучаем технологии будущего. [Электронный ресурс] URL: http://neuralnet.info/глава-1-введение/ (дата обращения: 22.02.2018).

27. Неочевидные возможности ABBYY FineReader. [Электронный ресурс] URL: https://habrahabr.ru/company/abbyy/blog/116542/ (дата обращения: 25.02.2018).

28. Репозиторий проекта на Bitbucket. [Электронный ресурс] URL: https://bitbucket.org/webdinislam/recognizemath/ (дата обращения: 24.03.2018).

29. Сверточные нейронные сети: вгляд изнутри – CodeSide. [Электронный ресурс] URL: http://ru.datasides.com/code/cnn-convolutional-neural-networks/ (дата обращения: 20.02.2018).

30. Чочиа П.А. Сегментация изображений на основе анализа расстояний в пространстве признаков. // Автометрия. – 2014. – Vol. 50. – No. 6. – P. 97–110.

ПРИЛОЖЕНИЯ

Приложение 1. Скриншоты веб-сервиса

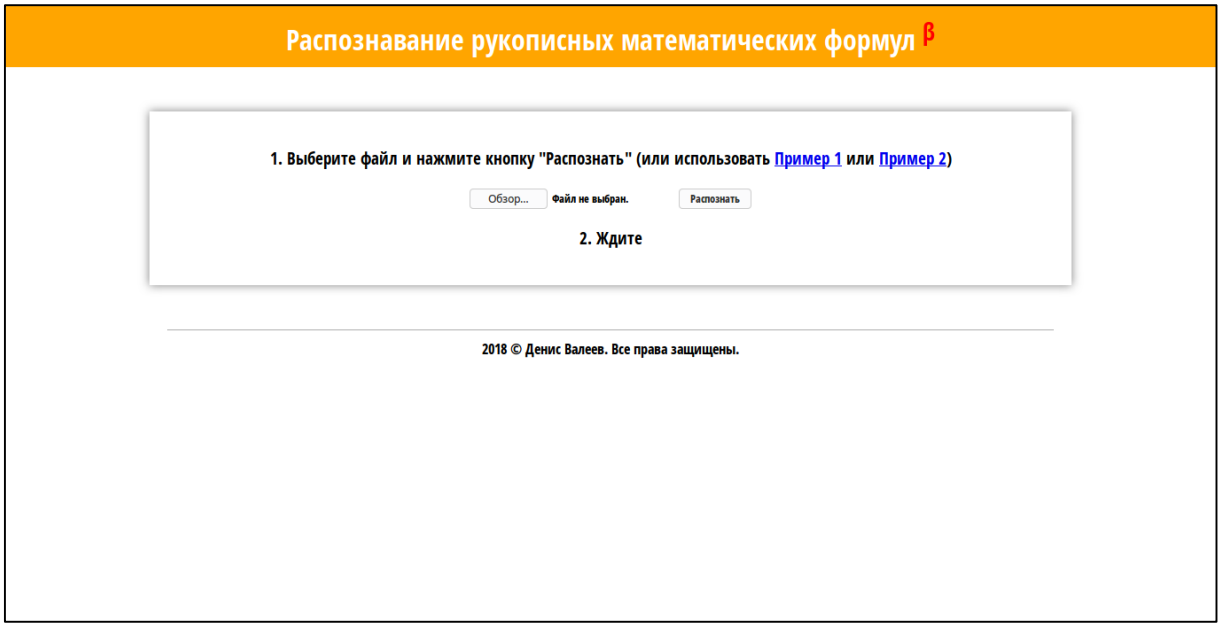

Рис. 1. Начало работы веб-сервиса

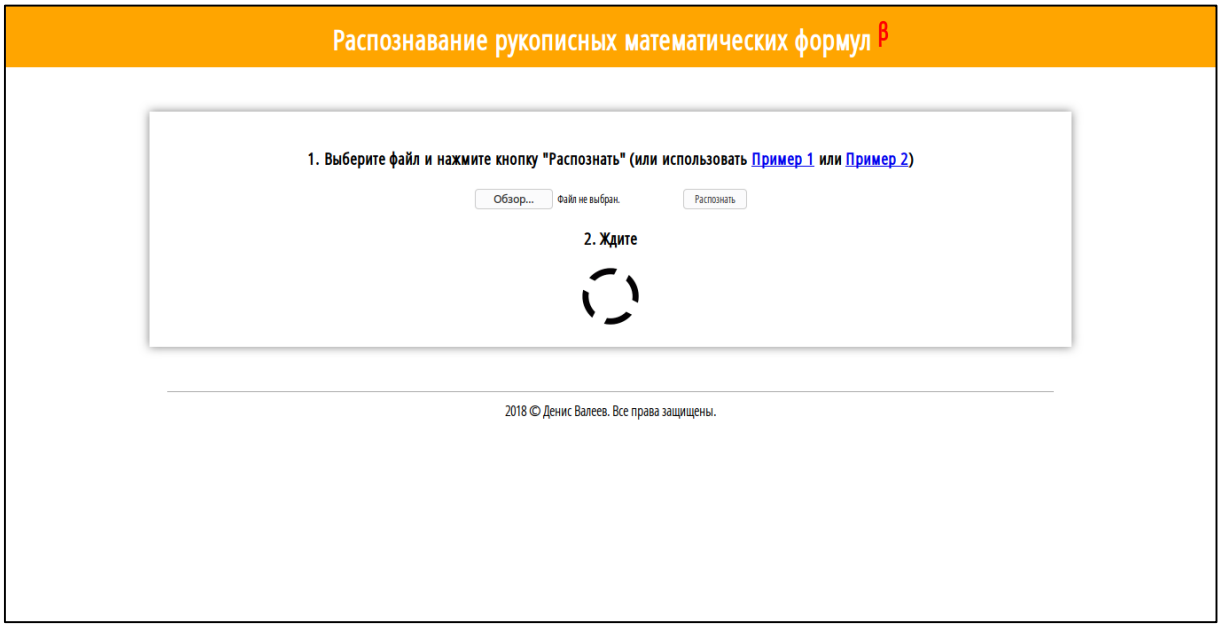

Рис. 2. Процесс распознавания

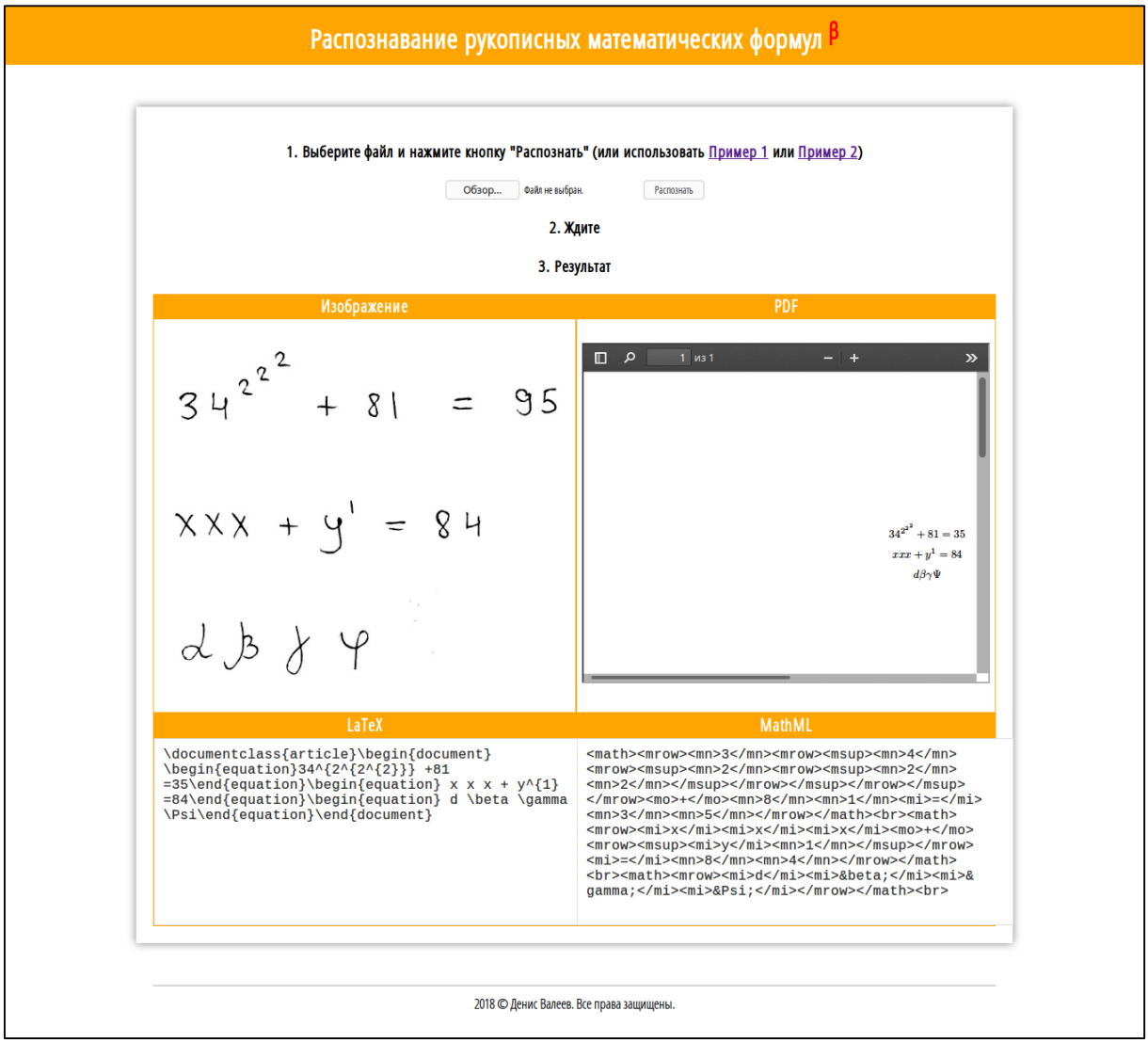

Рис. 3. Результат распознавания

Приложение 2. Примеры распознавания

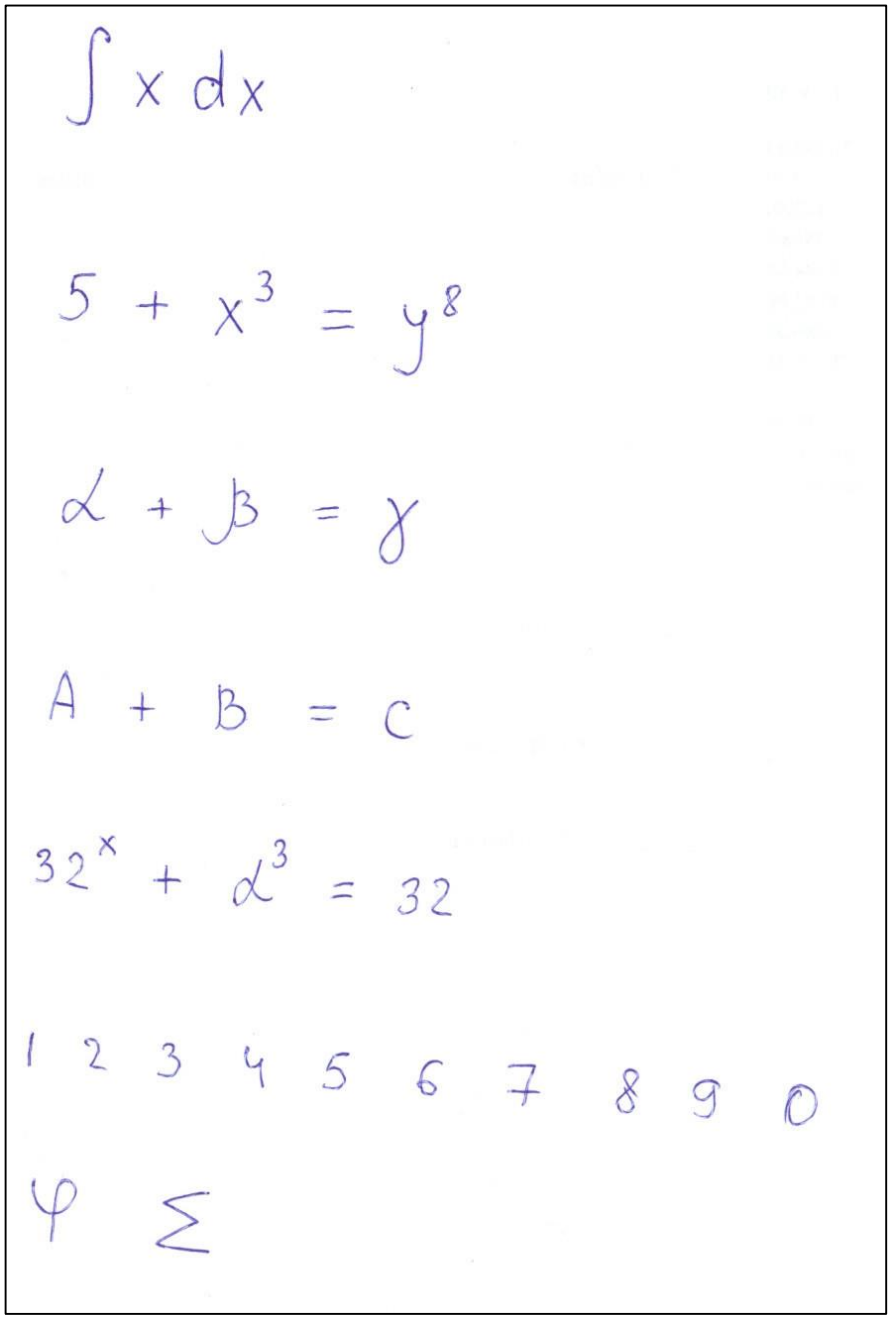

Рис. 4. Пример скана рукописных формул 1

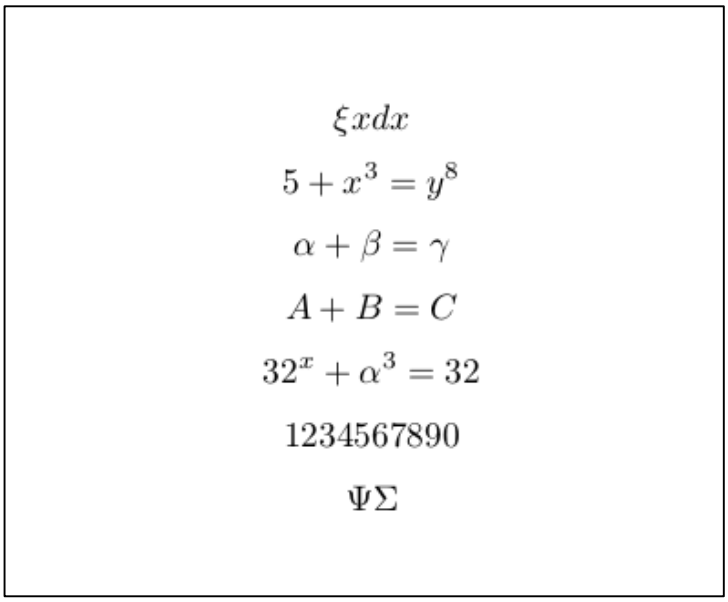

Рис. 5. Полученный результат примера 1

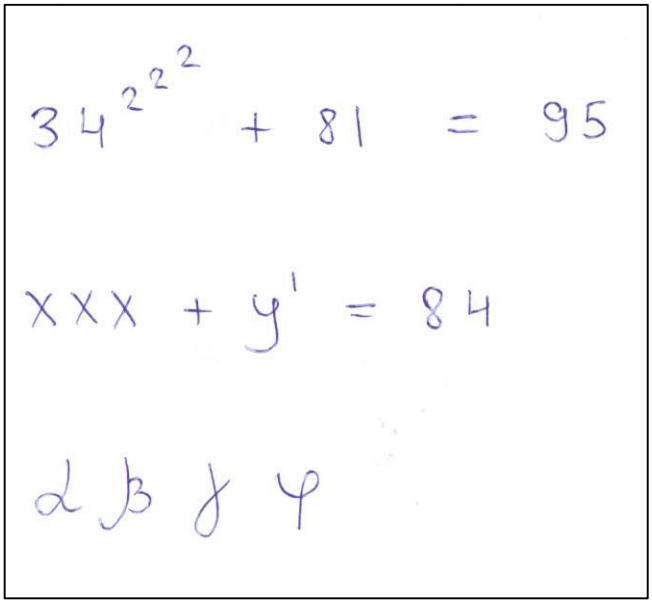

Рис. 6. Пример скана рукописных формул 2

$$34^{2^{2^2}} + 81 = 35 \quad (1)$$

$$xxx + y^1 = 84 \quad (2)$$

$$\alpha B \gamma \Psi \quad (3)$$

Рис. 7. Полученный результат примера 2

Приложение 3. Награды


РОССИЙСКАЯ НАУЧНО-СОЦИАЛЬНАЯ ПРОГРАММА ДЛЯ МОЛОДЕЖИ И ШКОЛЬНИКОВ

ШАГ В БУДУЩЕЕ

XIII Уральская выставка научно-технического творчества молодежи «Евразийские ворота России»

Региональный филиал Российской молодежной научной и инженерной выставки «Шаг в будущее»

(г. Челябинск, Южно-Уральский государственный университет, 2-4 апреля 2018 г.)

Д И П Л О М

ПЕРВОЙ СТЕПЕНИ

за лучшую творческую работу по информационным технологиям

с вручением памятной медали выставки

Н А Г Р А Ж Д А Е Т С Я

ВАЛЕЕВ Денис Ильфатович

г. Челябинск, ФГАОУ ВО «Южно-Уральский государственный университет» (НИУ), КЭ-401 (бакалавр)

Разработка системы обработки математических рукописных формул с применением нейросетевых технологий

Научный руководитель: Голодов Валентин Александрович, кандидат физ.-мат. наук, Южно-Уральский государственный университет

Ф.Л. Дегтярев
Президент
Южно-Уральской
торгово-промышленной
палаты, координатор
Уральской выставки НТТМ
«Евразийские ворота России»

А.А. Радионов
Проректор
по учебной работе
Южно-Уральского
государственного университета,
научный руководитель программы
«Шаг в будущее»

Е.Н. Кузьмин
Член Центрального совета
Всероссийской программы для
молодежи «Шаг в будущее»,
координатор Южно-Уральской
программы для молодежи «Шаг
в будущее-Созвездие-НТТМ»

4 апреля 2018 г.

Рис. 8. Диплом первой степени