Оглавление

[Аннотация 3](#_Toc159185575)

[Ключевые слова: 3](#_Toc159185576)

[Какие проблемы могут возникать у студентов во время лекций 3](#_Toc159185577)

[Кейс №1 Проблемы со слухом 3](#_Toc159185578)

[Кейс №2 Проблемы с зрением 3](#_Toc159185579)

[Кейс №3 Проблема потери контекста 4](#_Toc159185580)

[Кейс №4 Скорость написания материала преподавателя 4](#_Toc159185581)

[Кейс №5 Проблема стирания материала с доски 4](#_Toc159185582)

[Кейс №6 Проблема понимания почерка преподавателя (целого материала или части) 4](#_Toc159185583)

[Кейс №7 Отсутствие студента по уважительной причине 4](#_Toc159185584)

[Какие неприятные ситуации могут возникнуть если ввести помощника по лекциям 4](#_Toc159185585)

[Общие алгоритмы программного обеспечения и нейросетевой модели 5](#_Toc159185586)

[Получение данных 5](#_Toc159185587)

[Предварительная обработка 5](#_Toc159185588)

[Улучшение контрастности и яркости 6](#_Toc159185589)

[Фильтрация 7](#_Toc159185590)

[Бинаризация 7](#_Toc159185591)

[Коррекция изображений 7](#_Toc159185592)

[Устранение фонового шума 8](#_Toc159185593)

[Нормализация 8](#_Toc159185594)

[Сегментация 8](#_Toc159185595)

[Извлечение признаков 9](#_Toc159185596)

[Классификация 9](#_Toc159185597)

[Постобработка 9](#_Toc159185598)

[Используемые модели 9](#_Toc159185599)

[Convolutional Neural Networks (CNN) 9](#_Toc159185600)

[Recurrent neural network 10](#_Toc159185601)

[LSTM (BLSTM) 11](#_Toc159185602)

[MRF, HMM (MQDF) 11](#_Toc159185603)

[Резюмируя 12](#_Toc159185604)

[Датасеты 12](#_Toc159185605)

[LaTeX 16](#_Toc159185606)

[Решение задач 18](#_Toc159185607)

[State of Art 18](#_Toc159185608)

[Планируемый инструментарий для разработки модели и ПО, а также написания текста работы. 23](#_Toc159185609)

[Оценка полезности материалов представленных статей 23](#_Toc159185610)

[Дальнейшие планы по работе с материалом: 24](#_Toc159185611)

[Примерный план разработки приложения по распознаванию рукописных математических выражений (ответ ChatGPT 3.5) 24](#_Toc159185612)

[Топ 3 глупых вопросов ChatGPT 3.5 26](#_Toc159185613)

# Аннотация

В данной работе рассматриваются технологии, алгоритмы и методы разработки программного обеспечения (далее ПО), использующего машинное обучение для распознавания, решения и преобразования в LaTeX рукописных математических выражений.

В первом блоке данной работы рассматривается общий алгоритм и схема работы программного обеспечения. Рассматриваются структура нейросетевой модели, структура и архитектура программного обеспечения, используемые библиотеки и языки программирования.

Во втором блоке данной работы рассматривается подробное описание и роль различных нейросетевых моделей в функционировании программного обеспечения. Раскрываются типы моделей, их алгоритм, а также роль в ПО и воздействие на входящие данные. Также рассматриваются необходимые инструменты для разработки ПО.

В третьем блоке рассматриваются варианты датасетов которые могут быть использованы при разработке и обучении нейросетей. Рассматриваются виды датасетов, правовой статус, а также структура.

В четвертом блоке рассматривается возможность преобразование полученных данных в формат компьютерной верстки LaTeX. Данный формат широко распространен в задачах отображения и решения различного рода математических задач.

В заключительном блоке рассматривается возможность решения оцифрованных математических выражений.

## Ключевые слова:

Сверточные нейронные сети (CNN), Рекурентные нейронные сети (RNN), датасет, CROHME, MNIST, LaTeX, рукописные математические выражения, Python, Anaconda IDE, Tenserflow Keras, Interactive Python, MatPlotLib.

# Какие проблемы могут возникать у студентов во время лекций

Первые кейсы связаны с проблемами студентов

## Кейс №1 Проблемы со слухом

Существуют ситуации временной потери слуха у студентов (например, по причине воспаления ушного прохода при отите). Поэтому студент может плохо воспринимать лекционный материал и не понимать смысл представленных на доске формул. При распознавании формул, текста и речи преподавателя, студент может ознакомится с печатным вариантом лекции что повысит степень понимания материала.

## Кейс №2 Проблемы с зрением

Делюсь собственным опытом, бывали случаи, когда я по разным причинам забывал очки, получалась очень неприятная ситуация, связанная с тем, что часть лекций я распознавал хоть и с трудом, а часть вообще не понимал. Плюс приходилось очень сильно напрягать глаза и щуриться что очень неприятно как физически, так и психологически.

## Кейс №3 Проблема потери контекста

По различным причинам студенты могут отвлечься от потока лекции и потерять мысль и контекст которые транслирует преподаватель. Из-за этого начинается непонимание лекции и снижается уровень восприятия остального материала и возможно последующих лекций.

Следующие кейсы могут быть напрямую или косвенно связаны с преподавателем

## Кейс №4 Скорость написания материала преподавателя

Поскольку все мы разные, уровень скорости восприятия информации у нас разный. Существуют люди чья скорость выдачи информации (как устной, так и письменной) очень высока. И поспевать за ними бывает непросто, из-за этого может также возникать проблема, которая описана в кейсе №3.

## Кейс №5 Проблема стирания материала с доски

Проблема может быть косвенно связана с кейсом №4. Из-за скорости письма преподавателя доска может быстро заполнятся, из-за этого необходимо освобождать новое место стирая предыдущие записи. Поэтому теряется часть полезного материала, а также возникают проблемы так называемых «Незаконченных формул».

## Кейс №6 Проблема понимания почерка преподавателя (целого материала или части)

Рассмотрим ситуацию когда у преподавателя специфичный почерк, тогда возникают такие проблемы как, непонимания полного смысла лекционного материала или части текста. Из-за этого снижается понимание предметной области.

## Кейс №7 Отсутствие студента по уважительной причине

Введение подобной системы может решить проблему студентов пропустивших лекции по уважительной причине (например по состоянию здоровья или по семейным обстоятельствам).

## Какие неприятные ситуации могут возникнуть если ввести помощника по лекциям

На самом деле существует проблема (пришедшая в мою голову), которая может возникнуть если ввести подобного помощника так это лень студентов на паре и отказ писать лекции. А ведь и правда многие из нас по природе ленивы, и будут рассуждать так: «Зачем мне писать конспекты если есть система, которая за меня подготовит их». Также введение подобной системы может спровоцировать прогулы пар.

# Общие алгоритмы программного обеспечения и нейросетевой модели

Согласно данным из статьи Aman Chandra, «Handwritten Digit Recognition using Machine Learning» существуют две большие группы приложений, а именно онлайн и оффлайн распознавание математических выражений.

Первая группа не говорит о том, что приложение должно быть в формате Web, а лишь о том, что приложение отслеживает действие пользователя в режиме реального времени (чаще всего это можно представить в виде ввода математического выражения с помощью курсора или пера в специально отведенное пространство) и постоянно преобразовывает входные данные. То есть для такого ввода важен контекст. К примеру, когда пользователь ввел букву «О» то распознователь вывел эту букву, затем в нижней правой части буквы «О» он добавил «Черту» и буква «О» преобразовалась в букву «Q». Алгоритмы подобных приложений учитывают также такой параметр как ход пера в окне ввода. Такой метод отлично подходит для оцифровки рукописного ввода в печатный текст в режиме реального времени. Такой вид распознавания присутствует в относительно свежих редакциях Microsoft Office.

В свою очередь оффлайн распознавание подходит для распознавания символов, написанных на бумаге. Поскольку анализирует выражение целиком и сразу получает весь контекст. Процесс распознавания изображений можно поделить на несколько основных последовательных этапов. А именно:

1. Получение данных (Например в оффлайн варианте это загрузка изображения);
2. Далее идет предварительная обработка;
3. Затем «Сегментация» образов на изображении;
4. Следующим этапом будет «Извлечение признаков»;
5. Предпоследним этапом является «Классификация»;
6. И заключительным этапом «Постобработка»;

## Получение данных

На данном этапе пользователю предлагается, например загрузить изображение, для этого необходимо предоставить ему какой-либо элемент интерфейса в приложении. В качестве этого можно представить специально отведенную зону загрузки изображения, окно ввода выражения или диалоговое окно.

В качестве устройств ввода изображения может выступать камера смартфона или планшета, сканер изображений, перо с планшетом, или устройство сенсорного ввода без пера.

## Предварительная обработка

На данном этапе происходит подготовка изображений для дальнейшего анализа и включает в себя такие операции как:

1. Масштабирование изображения
2. Улучшение яркости и контрастности
3. Фильтрация
4. Бинаризация
5. Коррекция изображений
6. Устранение фонового шума
7. Нормализация

Масштабирование изображения является важной частью предварительной обработки изображении, **улучшение производительности алгоритмов:** поскольку некоторые алгоритмы могут работать более эффективно на небольших изображениях. Также **стандартизация размеров** может повысить качество обучения, поскольку многие алгоритмы предполагают наличие изображений с одинаковыми размерами. **Уменьшение вычислительной сложности:** Обработка больших изображений требует больших вычислительных вычислений. Масштабирование помогает уменьшить вычислительную сложность задачи, что может быть важным в условиях ограниченных ресурсов. **Борьба с вариабельностью данных:** Масштабирование помогает уменьшить вариабельность данных в наборе. Это может быть важным для моделей машинного обучения, чтобы лучше обобщаться на различные размеры объектов или различные условия изображений. **Уменьшение вычислительной сложности:** Масштабирование может помочь уменьшить влияние шумовых элементов или мелких деталей, делая изображение более чистым и легким для анализа.

Общий эффект масштабирования заключается в том, чтобы создать более удобные условия для последующего этапа анализа и распознавания, улучшая точность и эффективность алгоритмов.

### Улучшение контрастности и яркости

Улучшение контрастности и яркости изображения важно в контексте распознавания изображений по нескольким причинам:

**Выделение признаков:** Улучшение контрастности и яркости помогает выделить ключевые признаки и детали на изображении. Это особенно важно в случаях, когда объекты или текст на фоне изображения должны быть четко видны для эффективного распознавания. **Улучшение видимости текста:** В задачах распознавания текста, таких как распознавание рукописных выражений, увеличение контрастности и яркости помогает сделать текст более читаемым и легким для распознавания алгоритмами. **Уменьшение влияния теней и засветок:** Изображения могут содержать тени, засветки или неравномерное освещение, что затрудняет процесс распознавания. Улучшение контрастности и яркости может снизить влияние этих факторов, делая изображение более однородным. **Подготовка данных для алгоритмов:** Многие алгоритмы распознавания и машинного обучения предполагают, что данные подготовлены в определенном формате. Улучшение контрастности и яркости помогает стандартизировать изображения и создать более удобные условия для работы алгоритмов. **Снижение влияния шума:** Низкое качество изображений или наличие шумов может привести к ошибкам в распознавании. Улучшение контрастности и яркости может помочь снизить влияние шумов и повысить точность распознавания. **Адаптация к различным условиям освещения:** Улучшение контрастности и яркости также может помочь адаптироваться к различным условиям освещения, что важно для систем, работающих в различных средах.

Таким образом, улучшение контрастности и яркости играет ключевую роль в предварительной обработке изображений, создавая оптимальные условия для успешного распознавания объектов, текста и других важных деталей на изображении.

### Фильтрация

Фильтрация применяется для улучшения качества изображения и подготовки его для более эффективного анализа и распознавания. Вот несколько причин, почему фильтрация важна в задачах распознавания изображений:

**Снижение шума:** Изображения могут содержать шум, который представляет собой случайные или нежелательные детали, мешающие анализу. Фильтрация может помочь сгладить изображение и уменьшить влияние шума, делая изображение более чистым и подготовленным для распознавания. **Выделение границ:** Фильтры, такие как фильтры Собеля или фильтры Гаусса, могут быть использованы для выделения границ объектов на изображении. Это может быть полезно для последующего выделения объектов или текста в процессе распознавания. **Повышение контрастности:** Некоторые фильтры могут повысить контрастность изображения, что способствует лучшему выделению деталей и признаков. Это особенно важно для случаев, когда объекты имеют слабый контраст или находятся в условиях низкой освещенности. **Сглаживание текстур:** Фильтрация может помочь сгладить текстуры или детали, что может быть полезно при работе с изображениями, содержащими излишнюю детализацию. Это упрощает анализ и распознавание объектов. **Подавление помех:** В задачах распознавания объектов, особенно в условиях сложной среды, некоторые фильтры могут подавлять определенные виды помех, такие как мелкие объекты или части изображения, которые не являются интересующимися объектами.

Адаптация к особенностям изображения: Фильтрация может быть использована для адаптации к особенностям конкретных изображений, учитывая их свойства и особенности.

Общий эффект фильтрации заключается в улучшении качества изображения, уменьшении влияния шума и подготовке данных для более эффективного и точного распознавания объектов на изображении.

### Бинаризация

Процесс преобразования изображения в монохромное, где каждый пиксель относится либо к черному, либо к белому в зависимости от того проходит ли отдельный пиксель по порогу. На первом этапе исходя из выбранного порога пиксели с интенсивностью выше порога относят к одному классу (например, к классу объекта) а пиксели ниже порога к другому (например, к классу фона).

### Коррекция изображений

На этапе коррекции изображений исправляются такие недостатки как искаженная перспектива коррекция ориентации. А также исправление линейных искажений таких как сдвиг. Для устранения подобных искажений используется библиотека OpenCV 2.

### Устранение фонового шума

Для удаления лишних пикселей, несущих дополнительную информацию. Могут быть использованы различные фильтры основанные на математических или нейросетевых алгоритмах. Например, медианный фильтр заменяет значения пикселей на медианное, эффективен при устранении шума сохраняя контуры объекта.

Гауссовский фильтр сглаживает изображение, следовательно, уменьшает количество шума. Так же существует фильтры Собеля, Прюитта, Лапласа для выделения контура.

### Нормализация

Это процесс приведения изображений к определенным стандартам. Например, приведение пикселей к нулевому среднему и единичному стандартному отклонению. Также существует такое понятие как глобальная и локальная нормализации изображений. Глобальная нормализация может включать в себя масштабирование пикселей (Это будет полезно в случаях, когда интенсивность пикселей распределена равномерно), в свою очередь локальная нормализация ориентирована на отдельные участки изображения (к примеру алгоритм гистограммной эквализации – CLAHE который улучшает контраст в локальных зонах). В качестве последней операции нормализации изображений рассмотрим Аугментацию данных. Этот метод позволяет расширить выборку данных путем изменения формы, угла поворота, яркости и контраста исходного изображения.

## Сегментация

Процесс разделения образов на изображении на отдельные сегменты или регионы, которые представляют из себя объекты, структуры или области с определенными свойствами. Основная цель – выделение интересующих объектов на изображении, что упрощает их дальнейший анализ и обработку. Сегментация реализует такой важный принцип как декомпозиция сложносоставного до простых частей.

Применение сегментации может состоять из нескольких важных целей:

* Локализация объектов – это помогает определить границы и/или контуры объектов на изображении. Это важно при распознавании лиц, авто или других интересующих объектов
* Изоляция регионов изображения – это необходимо для выделения конкретной области, которая подлежит дальнейшему анализу. Например, определения на медицинских изображениях локализация определенных органов или тканей.
* Выделение и изоляция объектов с помощью сегментации может значительно улучшить точность алгоритмов распознавания и классификации.
* Сегментация как инструмент задания границ между объектами

Различают три вида сегментации исходя из их методов:

**Семантическая сегментация**, где каждому пикселю присваивается определенный класс;

**Инстанс-сегментация**, где каждый объект в сцене имеет уникальный идентификатор;

**Региональную сегментацию**, где изображение разбивается на регионы схожих цветов или текстур.

## Извлечение признаков

Извлечение признаков при распознавании представляет собой процесс выделения и представления ключевых характеристик или атрибутов из данных, которые затем могут быть использованы для обучения моделей машинного обучения, классификации, анализа данных или других задач. В контексте распознавания объектов, изображений или образов, извлечение признаков играет важную роль в создании репрезентаций, которые содержат информацию о существенных аспектах данных.

Использование предварительно обученных нейронных сетей для извлечения признаков из изображений или данных. Глубокие нейронные сети, такие как сверточные нейронные сети (CNN) или рекуррентные нейронные сети (RNN), способны автоматически извлекать высокоуровневые признаки из данных.

## Классификация

Классификация в контексте распознавания означает присвоение объектов, данных или образов к определенным категориям или классам на основе их характеристик. Это одна из ключевых задач в области машинного обучения и компьютерного зрения. Процесс классификации обучается на наборе данных, состоящем из примеров объектов, для определения правил и закономерностей, которые позволяют определить, к какому классу объект принадлежит.

## Постобработка

Постобработка в контексте распознавания обычно относится к этапу обработки результатов, полученных после применения алгоритмов распознавания или классификации. Этот этап может включать в себя различные техники и методы, направленные на улучшение качества, точности или интерпретируемости результатов.

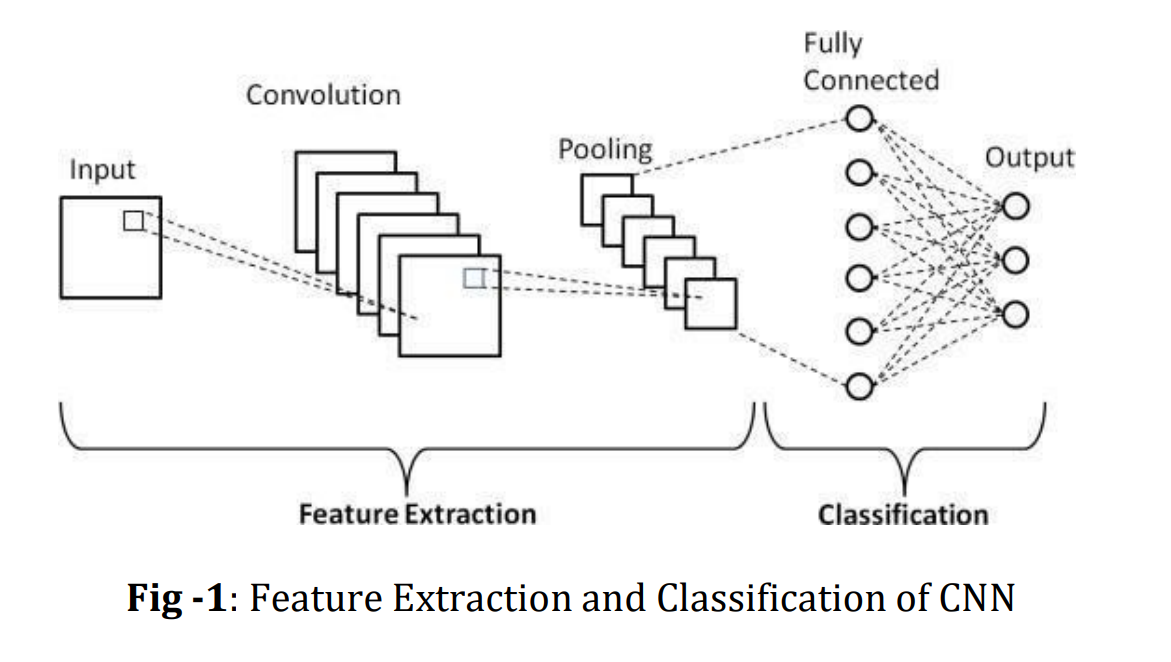
# Используемые модели

## Convolutional Neural Networks (CNN)

Самые распространенные модели в подобных классах задач это сверточные нейроные сети (CNN). Согласно статье Ankita Kawade «Handwritten equation solver using CNN» в качестве примера решателя квадратных уравнений используются CNN в представленной задаче модель используется как классификатор распознаваемых символов. Автор подмечает высокую эффективность глубоких нейронных сетей по сравнению с поверхностными, и в качестве примера называет многослойный прецептрон. В статье рассматривается метод онлайн распознавания математического выражения.

В тексте статьи также отражено примерное устройство данной модели, состоящей из следующих слоев:

|  |  |
| --- | --- |
| * Сверточный слой (Convolution Layer) – это самый основной слой любой CNN модели, на данном этапе изображение сканируется пиксель за пикселем, затем создается карта признаков для определения для будущих классификаций. * Слой подвыборки (Pooling Layer) – слой уплотнения информации, для включения только важной информации. | Извлечение признаков |
| * Полносвязный входной слой – это слой имеющий связь с каждым нейроном входного слоя. * Полносвязанные скрытые слои – слои в которых каждый нейрон соединен с каждым нейроном из предыдущего и последующего слоев. * Выходной слой - финальный слой модели CNN хранит результаты меток, определенных для классификации, и присваивает класс изображениям. | Классификация признаков |



## Recurrent neural network

Также нередко в задаче распознавания в совокупности с CNN (DenseNet) используют Рекуррентные нейронные сети (далее RNN). И согласно статье Guangcun Shan, «Robust Encoder-Decoder Learning Framework towards Offline Handwritten Mathematical Expression Recognition Based on Multi-Scale Deep Neural Network» рекуррентные нейронные сети используются в конвейере распознавания в качестве Декодера. **DenseNet (Densely Connected Convolutional Networks)** — это тип архитектуры сверточных нейронных сетей (CNN), предложенный в статье "Densely Connected Convolutional Networks" (2017) авторами Gao Huang, Zhuang Liu, Laurens van der Maaten, и Kilian Q. Weinberger. DenseNet представляет собой эволюцию сверточных нейронных сетей и предлагает улучшения в эффективности обучения и использовании параметров по сравнению с традиционными архитектурами, такими как VGG или ResNet.

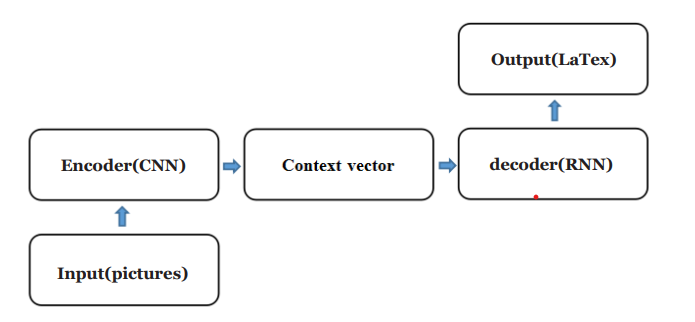


Схема конвейера

## LSTM (BLSTM)

LSTM или "Long Short-Term Memory" (Долгая краткосрочная память) это модель рекуррентной нейронной сети призванной решить так называемую проблему взрывного или затухающего градиента при обучении последовательных данных.

Одним из недостатков классических RNN моделей это проблема запоминания долгосрочных последовательностей и зависимостей, решается эта проблема путем добавления специальных конструкций так называемых «Врат» позволяющих сети контролировать поток данных. В свою очередь BLSTM означает "Bidirectional Long Short-Term Memory" (Двунаправленная долгая краткосрочная память) это RNN с Двумя LSTM слоями, один работающий в прямом направлении (от начала к концу последовательности) и другой в обратном направлений (от конца к началу последовательности).

## MRF, HMM (MQDF)

MRF (Markov Random Fields, Марковские случайные сети) могут быть использованы в методах онлайн распознавания рукописных символов в качестве инструмента сегментации и согласно статье Guangcun Shan, «Robust Encoder-Decoder Learning Framework towards Offline Handwritten Mathematical Expression Recognition Based on Multi-Scale Deep Neural Network» они используются в составе моделей RNN или LSTM для учета таких параметров как пространственные зависимости пикселей в каждом временном шаге.

Модель MRF позволяет предсказывать, согласно соседним пикселям, направление написания последующих частей единого символа.

HMM или "Hidden Markov Model" (Скрытая марковская модель) это статистическая модель которая используется для представления вероятностных последовательностей, особенно тех данных в которых состояние системы не наблюдается напрямую т.е. скрытые. В статье говорится, что в распознавании символов модель MRF зарекомендовала себя лучше чем HMM.

Метод MQDF обозначает "Mixture Quadratic Discriminant Function" (Функция смешанного квадратичного дискриминанта) это метод необходим для распределения данных на различные классы, и в качестве базы использует так называемые многомерные гауссовские распределения для каждого класса MQDF присваивает значение матрицы ковариации и среднее значение гауссовской компоненты.

Также существует модель LeNet-5 которая хорошо подходит для систем распознавания математических символов состоит из двух сверточных слоев, двух слоев подвыборки и одного полностью связанного слоя, и использует функцию SoftMax.

## Резюмируя

Согласно последним изысканиям для онлайн распознавания лучшим вариантом будет использование модели объединяющую многомасштабную сверточную нейронную сеть (CNN) с рекуррентной нейронной сетью с эффектом памяти (RNN) для идентификации двумерных рукописных математических выражений в виде одномерных последовательностей LaTeX.

Цитата «Мы усиливаем онлайн-кодировщик, используя архитектуру CNN-RNN»

В свою очередь для оффлайн хорошо подойдет кодировщик с использованием улучшенной CNN архитектуры DenseNet.

В статье «ICDAR 2019 CROHME + TFD: Конкурс по распознаванию рукописных математических выражений и обнаружению формул в наборе символов.» есть подробные любопытные сведенья об опыте команд участвующих в конкурсе.

# Датасеты

Согласно статьям:

* Aman Chandra, «Распознавание рукописных цифр с использованием машинного обучения: обзор»
* Chen Shan Wei, «Модель распознавания рукописных чисел на основе сверточной нейронной сети для четырех арифметических операций»
* Mohini Lokhande, «Решатель математических задач по рукописному вводу с использованием сверточной нейронной сети»
* Rajwardhan Shinde, «Решатель рукописных математических уравнений»

Для распознавания простых математических выражений можно использовать датасеты NIST (National Institute of Standards and Technology), MNIST (Modified National Institute of Standards and Technology) или EMNIST (Extended Modified National Institute of Standards and Technology).

Набор данных NIST это обширный набор данных содержащая в себе рукописные цифры и буквы, лица и печатные документы. Различные поднаборы NIST могут использоваться для различных задач, включая распознавание текста, распознавание образов лиц и другие.

MNIST это модифицированный набор данных NIST с добавленными тренировочными изображениями в количестве 60 тысяч и 10 тысяч текстовых изображений размером 28 на 28 px.

EMNIST – это расширенный модифицированный набор данных включающий в себя латинский алфавит в нижнем и верхнем регистре.

В свою очередь в статьях:

* Guangcun Shan, «Устойчивая обучающая структура энкодера-декодера для распознавания оффлайн рукописных математических выражений на основе многомасштабной глубокой нейронной сети»
* Hai Dai Nguyen, «Глубокие нейронные сети для распознавания онлайн рукописных математических символов»
* Mahshad Mahdavi, «Конкурс по распознаванию рукописных математических выражений и обнаружению формул в наборе символов»
* Masaki Nakagawa, «Распознавание онлайн рукописных математических символов с использованием глубоких нейронных сетей»
* Pooja Chaudhari, «Распознавание рукописных математических выражений с использованием агрегации внимания на основе взаимного обучения в обоих направлениях»

Были описаны процессы разработки моделей с использованием наборов данных CROHME «Competition on Recognition of Online Handwritten Mathematical Expressions» это конкурс, созданный для стимулирования исследований и разработок в области распознавания онлайн рукописных математических выражений. Конкурс был создан с целью продвижения технологий распознавания рукописных математических формул и выражений, что имеет практическое значение для автоматизации систем математического распознавания, создания образовательных инструментов и других приложений.

Для разработки приложений был создан датасет CROHME 2011 и с проведением каждого нового конкурса датасет пополняется количество тренировочных данных, последний конкурс был в 2019 году.

Согласно официальным данным состав датасета представлен в таблице ниже:

|  |  |  |
| --- | --- | --- |
| **Название** | **Кол-во тренировочных** | **Кол-во тестовых** |
| CROHME 2011 | 180 440 | 4 580 |
| CROHME 2012 | 5 250 |
| CROHME 2013 | 6 080 |
| CROHME 2014 | 9 999 |

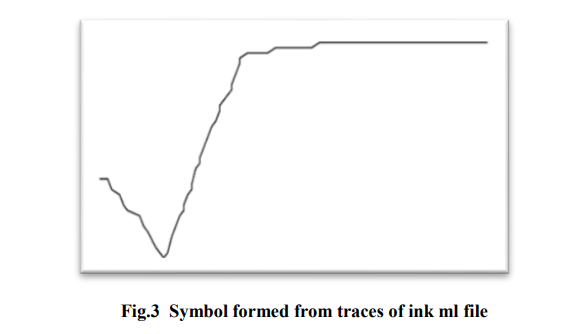
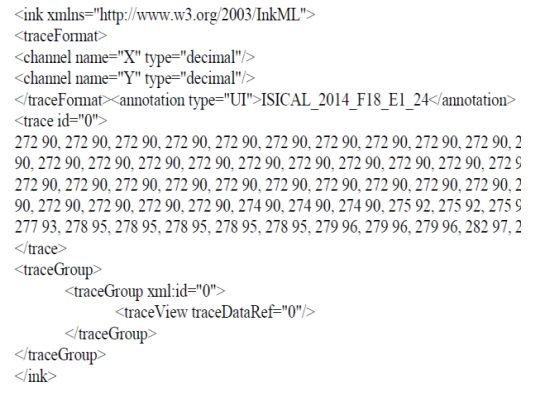
Что же представляют из себя данные датасета CHROME:

Цитата из статьи:

Мы использовали набор данных математических символов, предоставленный CROHME (Конкурс по распознаванию онлайн-рукописных математических выражений). Этот набор данных представляет собой коллекцию рукописных математических символов, которые могут быть использованы для формирования математических выражений [3].

Набор данных состоит из файлов ink ml, содержащих следы математических символов, собранные при вводе данных несколькими пользователями по всему миру. Этот набор данных используется для обучения модели нейронной сети, используемой для распознавания математических выражений.

Этот набор данных также предоставляет файл с истиной, в котором перечислены все файлы ink ml, сопоставленные символам, которые они представляют. Наш основной фокус будет на файлах ink ml, и файл истины используется для сопоставления следов их символам, которые используются для распознавания математического символа. Файл ink ml содержит следы математических символов, как показано ниже:



**Fig. 2 Sample ink ml file of CROHME dataset**

По типам данных в датасете CROHME согласно статье «Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection» от автора Mahshad Mahdavi можно разделить на следующие типы:

* Формулы
* Символы
* Структуры

Также существует следующие датасеты не представленные в статьях, но отраженные в других источниках сети интернет:

**CROHME (Competition on Recognition of Online Handwritten Mathematical Expressions):**

Тип данных: CROHME представляет собой набор данных, специально созданный для конкурса по распознаванию онлайн рукописных математических выражений.

Размер: Размер может варьироваться в зависимости от конкретной версии CROHME.

Использование: Датасет CROHME содержит рукописные математические выражения с разнообразными символами и структурами, что делает его полезным для обучения моделей распознавания математических формул.

**InftyDB (Infty Document Database):**

Тип данных: InftyDB содержит рукописные и распечатанные математические формулы, представленные в виде документов.

Размер: Включает в себя тысячи документов с математическим содержанием.

Использование: Этот датасет полезен для задач, связанных с распознаванием и анализом математических выражений в текстовых документах.

**MfrDB (Math Formula Recognition Database):**

Тип данных: MfrDB предоставляет изображения рукописных математических формул с различными символами и структурами.

Размер: Размер может быть значительным, включая тысячи изображений формул.

Использование: MfrDB предназначен для задач распознавания и классификации рукописных математических формул.

**IAM Online Handwriting Database:**

Тип данных: IAM содержит рукописные тексты, включая математические выражения, написанные от руки.

Размер: Большой набор данных с разнообразными текстами и символами.

Использование: Полезен для обучения моделей распознавания рукописных символов и выражений, включая математические.

**OpenAI's Math Dataset:**

Тип данных: Этот датасет содержит математические уравнения и выражения, сгенерированные с использованием искусственного интеллекта.

Размер: Размер может быть значительным, и он предоставляет разнообразные математические контексты.

Использование: Может быть использован для задач генерации и распознавания математических выражений.

**CROHME (Competition on Recognition of Online Handwritten Mathematical Expressions):**

CROHME может быть доступен для бесплатного использования в рамках конкурса. Рекомендуется проверить официальный веб-сайт конкурса или связанные ресурсы для получения подробной информации.

**InftyDB (Infty Document Database):**

Информация о доступности InftyDB для бесплатного использования может зависеть от конкретных условий предоставления данных. Рекомендуется проверить официальный источник или ресурсы, связанные с базой данных.

**MfrDB (Math Formula Recognition Database):**

|  |  |  |
| --- | --- | --- |
|  | **Test** | **Train** |
| Number of documents | 36 | 10 |
| Number of pages | 569 | 236 |
| Number of single-character math expressions | 7506 | 2556 |
| Number of multi-character math expressions | 18890 | 9329 |
| Total Number of math expressions | 26396 | 11885 |

Информация о бесплатном доступе к MfrDB может зависеть от политики создателей базы данных. Рекомендуется проверить их официальный источник или ресурсы.

**IAM Online Handwriting Database:**

IAM Online Handwriting Database доступен для бесплатного использования в академических исследованиях. Условия использования могут варьироваться, но обычно он распространяется с открытой лицензией.

**OpenAI's Math Dataset:**

Информация о доступности Math Dataset от OpenAI может быть предоставлена на официальном веб-сайте OpenAI. OpenAI обычно поддерживает политику открытого доступа к своим данным.

# LaTeX

Латех это язык и система компьютерной верстки, предназначенная для создания документов, математических формул, верстки книг, рендера математических формул на веб страницах, является официально поддерживаемым в стандарте языка веб документов Markdown и т.д.

Приложения, основанные на языке Латех используются в качестве редактора документов и существуют в виде десктоп и веб приложений, встроены во многие сервисы, такие как системы управления репозиториями GitHab и GitLab.

Поскольку является общепризнанным и широко распространенным часто используется в системах распознавания рукописных математических формул и текстовых документов.

Статьи затрагивающие эту тему:

* «Recognition of On-Line Handwritten Mathematical Expressions in the E-Chalk System - An Extension» от автора Ernesto Tapia из Берлина;
* «Robust Encoder-Decoder Learning Framework towards Offline Handwritten Mathematical Expression Recognition Based on Multi-Scale Deep Neural Network» от автора Guangcun Shan из Пекина;
* «Converting math formula images to LaTeX encoding with machine learning» от Joseph Salmon из Монпелье (Франция);
* «Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection» от Mahshad Mahdavi из США;
* «Application of Methods of Machine Learning for the Recognition of Mathematical Expressions» от Олега Вереса из Украины;
* «Recognizing handwritten mathematical expressions» от Pallavi Ratra из Индии;
* «Handwritten Mathematical Equations Conversion to LaTeX Equivalent» от Pooja Chaudhari из Индии;
* System for Recognition and Evaluation of Handwritten Arithmetic Expressions от Vlado Galic из Хорватии.

Исходя из данных отраженных в статьях, LaTeX во многих задачах по распознаванию математических символов и выражений будет конечным форматом ответа, готовым к отображению, а также возможностью решения полученного выражения или уравнения.

E-Chalk например является системой для распознавания лекционных материалов вводимых на интерактивных досках. В своей основе использует модель Support Vector Machine (SVM). Написана на языке Java. Преобразует данные в формат LaTeX и MathML. Имеет на борту встроенный решатель уравнений.

В задачах от конкурса CROHME четко указано что приоритетными данными для результата преобразования являются форматы Latex, MathML и LG. Далее данные конвертируются и отправляются в SymLG для анализа корректности распознавания, выявления ошибок классификации.

Олег Верес реализовал приложение с окном ввода математического выражения на языке Java используя Фреймворк Weka и библиотеку JLatexMath.

В статье Pallavi Ratra сказано что часто для преобразования двумерных математических выражений в одномерные последовательности Latex используются кодировщик основанный на CNN в качестве наблюдателя и рекуррентная нейронная сеть с механизмом внимания для декодирования и генерации вывода в нужный для результата формат (Latex).

Цитата из статьи «Например, Latex и MathML требуют знания предопределенных наборов ключевых слов для описания специальных математических символов и функций, а также пространственных композиций. Другие инструменты, такие как Math Type, зависят от визуальной среды для добавления символов с использованием мыши и требуют много времени.»

Используется метод с принципами Watch, Append and Parse, который представляет собой новый комплексный подход на основе нейронной сети.

**Watch (Следить)**

**Append (Добавить)**

**Parse (Разбор)**

# Решение задач

После классификации задачи существует возможность встроить решатели для получения готовых ответов. Однако существует и другой поход – прямой:

Согласно статье «Recognition and Solution of Handwritten Partial Differential Equations» от автора Ahmed Yehia из Египта рассматривает возможность решения дифференциальных уравнений в частных производных, для решения задач экологии, а именно загрязнения. В своей работе автор рассматривает алгоритм “Nangs”.

Автор рассматривает возможность решения Дифф. Уравнений без процесса дискретизации, а непосредственно с помощью глубокого обучения. Также в статье было сказано что помимо алгоритма “Nangs” существуют алгоритмы PyDEns и NeyroDiffEq.

В качестве датасета рассмотрены MNIST, CROHME а также собственный датасет с 20 000 данных. И лучший результат показал последний вариант, процент успешных результатов 97,1%. Также отмечается что лучшим алгоритмом решения является «Nangs», это алгоритм построенный на основе библиотеки PyTorch.

Также существует более простой метод для более легких задач, а именно распознавание символов, преобразование в математические выражения и решение в запрограммированных алгоритмах.

Подобный метод описан в статье «Ручной решатель уравнений с использованием сверточных нейронных сетей (CNN).» от автора Ankita Kawade из Индии, в работе рассматривается решение таких задач как квадратные уравнения.

# State of Art

Из систем по распознаванию математических выражений можно особенно выделить 2 приложения, а именно «MathPix Snipping Tool» и «PhotoMath».

Первый представляет собой систему по распознаванию рукописных выражений и конвертатор результата в форматы LaTeX, KaTeX и MathML.

В приложений существует возможность распознавания как с помощью начертания в специальном поле математических выражений, так и распознавания с фотографий.

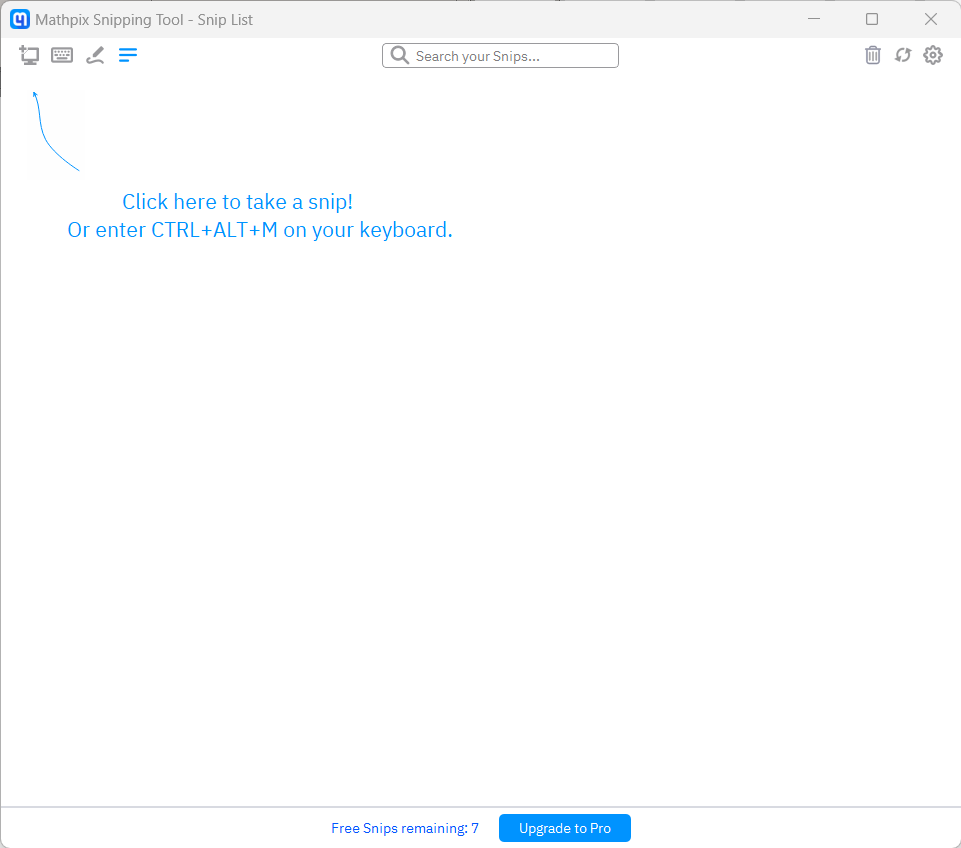
В свою очередь «PhotoMath» это удобный решатель математических примеров полученных со снимков учебников или рукописных выражений.

Давайте подробнее рассмотрим эти инструменты и начнем с «MathPix Snipping Tool»:

Основное окно программы представляет собой пустое окно сверху и слева которого расположены

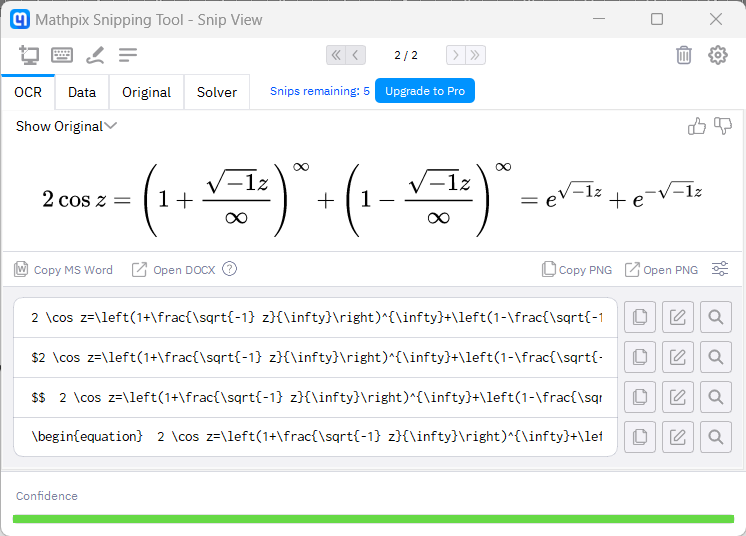
Иконки нажав на которые мы можем выбрать способы распознавания

Первый вариант позволяет создать скриншоты из документов которые необходимо распознать



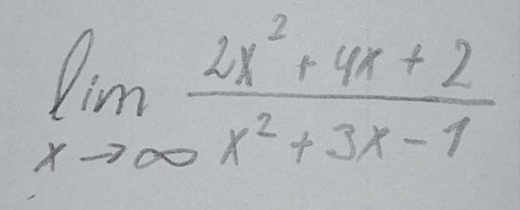
Приведу примеры распознавания разных условий:

Первый вариант, это распознавание на сайте или в веб-документе, я сделал скриншот со статьи об математическом анализе из википедии, и программа легко справилась. Предоставив разные варианты записи LaTeX, а именно классический LaTeX, 2 варианта для языка Markdown для написания выражения внутри текста с одним знаком доллара, и для написания выражения в отдельном блоке и посередине с двумя знаками доллара, и последний вариант это расширенный современный LaTeX. Также есть возможность скопировать MathML вариант для Microsoft Word и в виде PNG изображения.



Второй вариант будет представлять из себя выражение написанное от руки:

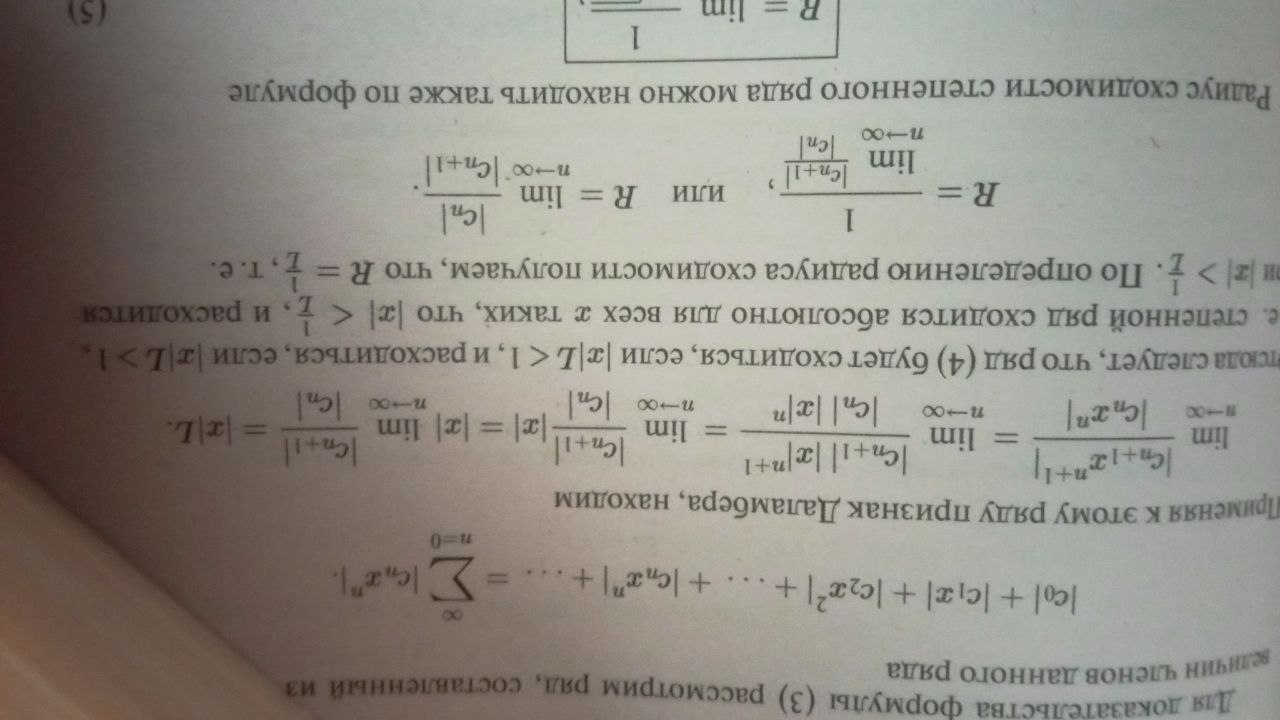
И в этом случае программа справилась на ура

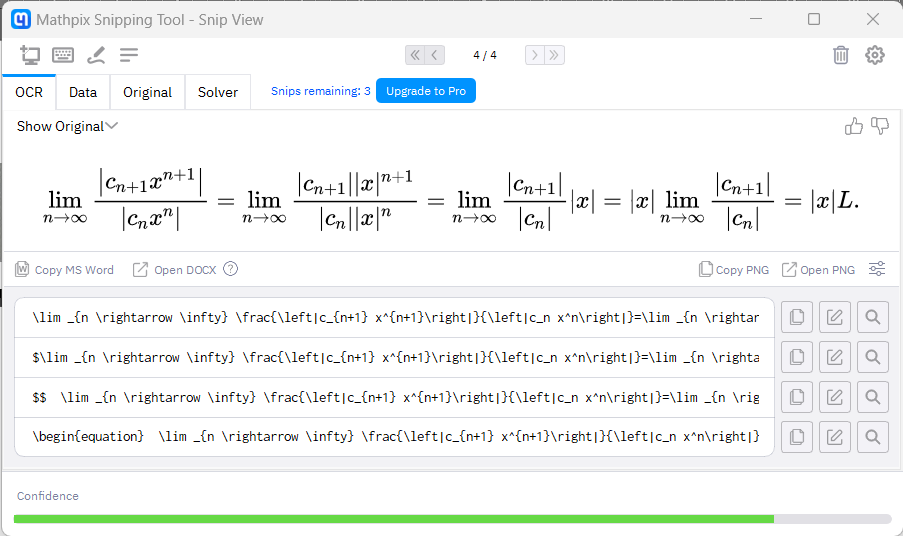


|  |  |
| --- | --- |
|  |  |

Также в программу встроен решатель примеров

И последним кейсом распознавания будет фотография книги плохого качества, в кадре присутствует затенение, есть нечеткие обрасти и фото перевернуто. Посмотрим как справится программа.



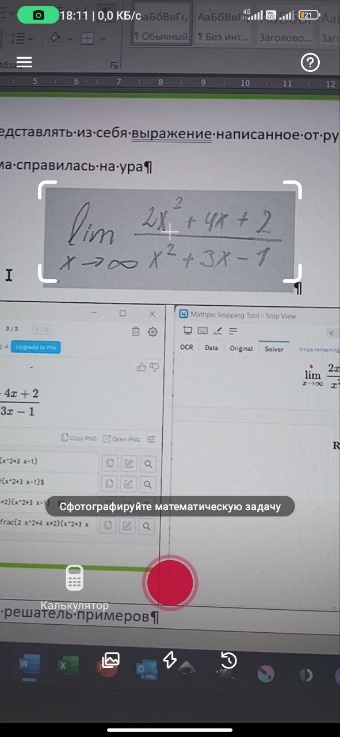


И программа справляется со своей задачей.

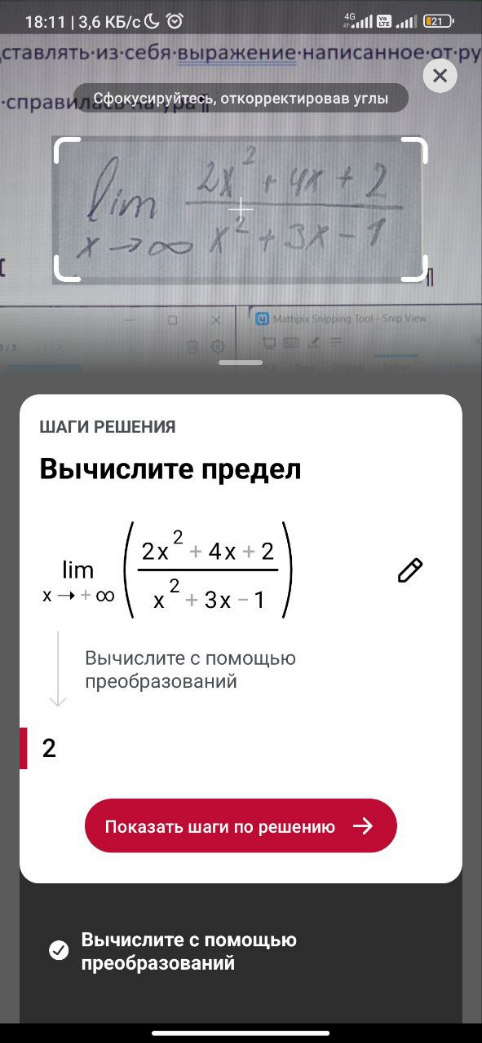
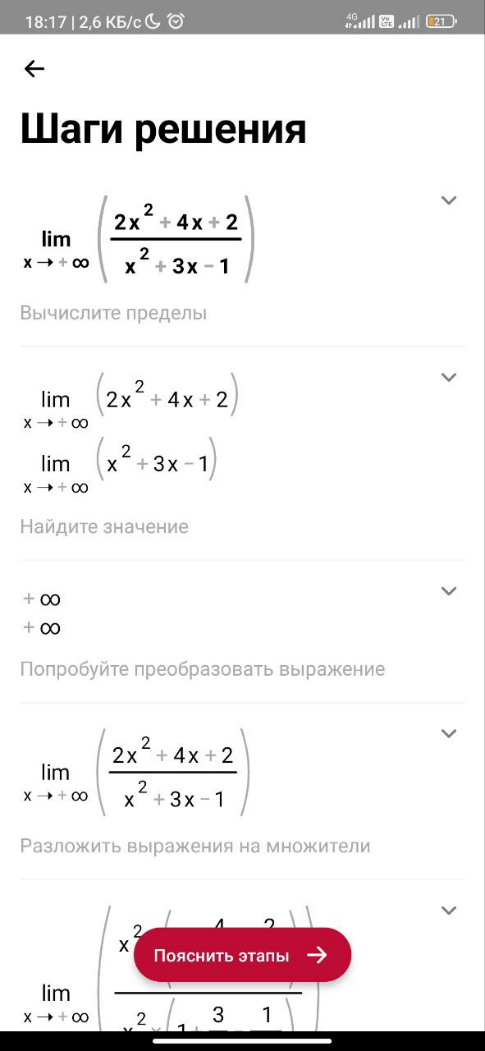
Давайте рассмотрим второй вариант (PhotoMath) и для разнообразия будем работать на смартфоне под операционной системой Андройд:

Я бы назвал это приложение для тех кто не любит решать математику или его можно использовать для самопроверки.

Программа выглядит следующим образом:



Это по сути камера с рамкой в которую нужно поместить распознаваемое выражение.

Отмечу что выражение распозналось корректно и сразу же приложение вывело результат.

Далее можно узнать шаги решения и получить пояснения по этапам решения задачи.

Выводы: В мире существуют множество подобны приложений, например Microsoft Math Solver как аналог PhotoMath, или xFormula как аналог MathPix Snip. Это в сою очередь подчеркивает необходимость в различных приложениях по распознаванию рукописных (или не очень) формул.

# Планируемый инструментарий для разработки модели и ПО, а также написания текста работы.

В качестве основного ин6стумена для прототипирования нейросетей предлагается использовать онлайн IPython редактор «Google Colab» и десктоп приложение – дистрибутив библиотек и языков, объединённых проблематикой машинного обучения и науки о данных «Anaconda»

В качестве редактора кода предлагается использовать Visual Studio Code поскольку он хорошо поддерживается сообществом, у него множество плагинов, он кроссплатформенный и самое главное он бесплатный.

Основным языком программирования будет Python 3 поскольку этот язык особо популярен в областях науки о данных и машинного обучения. В качестве основных библиотек рассматриваются PyTorch, Tenserflow Keras или Scikit-lern (необходимо уточнить данные из официальной документации о возможностях библиотек).

Также рассматривается возможность использования библиотеки OpenCV для детекции вводимых объектов, а также применения встроенных фильтров для работы с изображениями.

В качестве вопросно-ответных систем рассматриваются сервисы ChatGPT, YandexGPT и GigaChat ответы которых будут также подвергнуты анализу и сравнению полученных данных.

# Оценка полезности материалов представленных статей

Среди всего переведенного материала можно особо выделить следующие статьи:

* Ahmed Yehia, Recognition and Solution of Handwritten Partial Differential Equation Using Deep Learning
* Guangcun Shan, Robust Encoder-Decoder Learning Framework towards Offline Handwritten Mathematical Expression Recognition Based on Multi-Scale Deep Neural Network
* Hai Dai Nguyen, Deep Neural Networks for Recognizing Online Handwritten Mathematical Symbols
* Mahshad Mahdavi, Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection
* Masaki Nakagawa, Recognition of Online Handwritten Math Symbols using Deep Neural Networks
* Pallavi Ratra, RECOGNIZING HANDWRITTEN MATHEMATICAL EXPRESSIONS
* Pooja Chaudhari, Handwritten Mathematical Equations Conversion to LaTeX Equivalent
* Vlado Galic, System for Recognition and Evaluation of Handwritten Arithmetic Expressions

# Дальнейшие планы по работе с материалом:

* Перевести оставшиеся 20+ статей
* Перевод найденных крупных монографий
* Не прекращать поиски материалов
* Приступить к созданию модели
* Приступить к проведению эксперимента
* Работать в поте лица и головой

## Примерный план разработки приложения по распознаванию рукописных математических выражений (ответ ChatGPT 3.5)

Разработка веб-приложения для распознавания математических выражений с использованием алгоритмов машинного обучения может быть разделена на несколько этапов. Вот примерный план разработки:

**1. Определение требований:**

- Определите основные цели вашего приложения.

- Определите форматы входных данных (фотографии, рукописные записи, текстовые выражения).

**2. Исследование и выбор алгоритмов:**

- Изучите доступные алгоритмы машинного обучения для распознавания текста и символов.

- Выберите подходящие алгоритмы для вашего контекста, такие как сверточные нейронные сети (CNN) или рекуррентные нейронные сети (RNN).

**3. Подготовка данных:**

- Соберите и подготовьте набор данных, содержащий различные примеры математических выражений.

- Разделите данные на обучающий и тестовый наборы.

**4. Обучение модели:**

- Разработайте модель, используя выбранные алгоритмы машинного обучения.

- Обучите модель на вашем обучающем наборе данных.

- Оцените производительность модели на тестовом наборе.

**5. Интеграция с веб-приложением:**

- Разработайте бэкенд-часть приложения для взаимодействия с моделью.

- Создайте API для отправки данных на обработку и получения результатов.

- Интегрируйте обученную модель в ваше веб-приложение.

**6. Разработка пользовательского интерфейса:**

- Создайте пользовательский интерфейс для ввода математических выражений.

- Реализуйте функционал загрузки изображений или ввода текста.

**7. Тестирование:**

- Проведите тестирование приложения, уделяя внимание точности распознавания и производительности.

- Разработайте автоматические тесты для проверки работоспособности.

**8. Оптимизация и масштабирование:**

- Оптимизируйте производительность модели и приложения.

- Рассмотрите возможность масштабирования для обработки большого числа запросов.

**9. Внедрение и поддержка:**

- Разверните приложение на сервере.

- Обеспечьте мониторинг и поддержку приложения в процессе эксплуатации.

- Рассмотрите возможность обновлений и доработок на основе обратной связи пользователей.

**10. Документация:**

- Создайте документацию для пользователя и разработчиков по использованию приложения и его API.

Этот план представляет общий подход к разработке веб-приложения для распознавания математических выражений. Реальные шаги могут варьироваться в зависимости от конкретных требований проекта и используемых технологий.

# Топ 3 глупых вопросов ChatGPT 3.5

