**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc184575065)

[**ГЛАВА 1** 4](#_Toc184575066)

[**1.1 Предмет анализа** 4](#_Toc184575067)

[**1.2 Машинное зрение** 4](#_Toc184575068)

[**1.3 Генерация текста** 5](#_Toc184575069)

[**1.4 Классические подходы решения проблемы базируются на 3-х шагах:** 5](#_Toc184575070)

[**1.5 Где может использоваться машинное зрение и генерация текста?** 7](#_Toc184575071)

[**ГЛАВА 2** 9](#_Toc184575072)

[**2.1 В ходе анализа данных по теме image captioning выяснилось:** 9](#_Toc184575073)

[**2.2 Для этого будут использованы ключевые библиотеки:** 9](#_Toc184575074)

[**2.3 Для реализации работы функционального GUI необходимо:** 10](#_Toc184575075)

[**2.4 Общий функционал должен включать:** 11](#_Toc184575076)

[**ГЛАВА 3** 14](#_Toc184575077)

# **ВВЕДЕНИЕ**

**Актуальность** обоснована ростом визуального контента, чей спад не будет наблюдаться в ближайшее время. Анализ большого количества данных стал рутинным и сложным делом. Для такого необходим инструментарий в виде приложения для обработки визуального контента и его точного описания. Данные технологии необходимы во множестве отраслях, науке и сфере услуг и развлечений.

Технология image captioning широко распространена в современном мире. Без неё не уже сложно представить жизнь любого человека даже в моменты, когда напрямую не пользуется ей, так как технология является полезным и важным инструментом.

**Цель:** разработка приложения для автоматической генерации текстового описания изображения.

**Объект исследования:** нейросети, машинное зрение, методология работы с генерацией естественного текста.

**Предмет исследования:** технология автоматической генерации текста, технология машинного зрения, технология автоматической генерации текста по изображению.

**Задача:**

- изучение предметной области в сфере автоматической генерации текстового описания изображения.

- анализ уже существующих решений.

- создание приложения для автоматической генерации текста по изображению.

# **ГЛАВА 1**

## **1.1 Предмет анализа**

Автоматическая генерация текстового описания изображения (Image Captioning) – технология описания изображения посредством методов машинного зрения.

Дальнейшая теория основана на ресурсе Papers with Code и его широкой базе данных различных исследований и статей в области машинного обучения. В рамках курсовой работы будут взяты статьи по теме Image Captioning и «наборов с перекрёстными описаниями COCO (Crisscrossed Captions, CxC).

## **1.2 Машинное зрение**

Одно из первых упоминаний для широких масс людей технология получила 14 апреля 2015 года вместе с выходом статьи «From Captions to Visual Concepts and Back», спонсированная Microsoft Research (MSR), что специализируется на исследованиях в области информатики. Текст статьи подробно раскрывает концепцию и логику работы машинного зрения. Это было важным витком развития, так как технология стала активно развиваться и интересовать не только научных деятелей, но и общественность.

Основная цель для создания и изучения машинного зрения – получения ключевых слов, которые характеризуют изображение. А именно процесс проходит следующим образом:

- **Извлечение признаков:** Машинное зрение позволяет извлекать важные признаки из изображений, такие как цвета, текстуры, формы и объекты. Эти признаки могут быть использованы для дальнейшего анализа и кластеризации.

- **Обнаружение объектов:** С помощью алгоритмов обнаружения объектов (например, YOLO, Faster R-CNN) можно идентифицировать и классифицировать объекты на изображении. Каждое обнаруженное событие может быть связано с ключевым словом.

- **Классификация изображений:** Модели глубокого обучения, такие как сверточные нейронные сети (CNN), могут быть обучены для классификации изображений по категориям. Каждая категория может быть представлена набором ключевых слов.

## **1.3 Генерация текста**

Данная статья ссылается на исследования UMIACS в другой статье «Corpus-Guided Sentence Generation of Natural Images», в которой подробно описан процесс автоматической генерации текста (Speech/Text Generation).

Серьёзной проблемой для генерации текста всегда является Natural language generation (NLG) или же естественное понимание языка. При генерации текста важно не только определить ключевые слова, но и составить в логическую правильную последовательность с соблюдением норм любого человеческого языка.

## **1.4 Классические подходы решения проблемы базируются на 3-х шагах:**

- выбор;

- планирование;

- реализация;

Точнее всего первый шаг изображён в самой статье   
в виде таблицы (рис. 1)

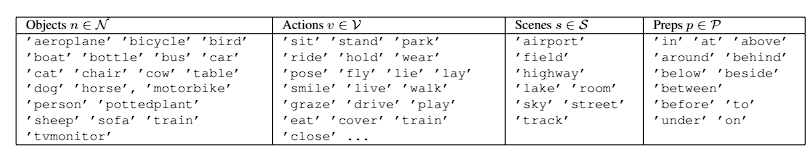


Рис. 1 Выбор подходящих слов под генерацию

Для описания множества образцов изображений необходимо понимать основные переменные, из которых и будет складываться строка описания:

- Объекты на изображении;

- Действия, которые происходят на изображении;

- Окружение сцены и самих объектов;

- Грамматически необходимые слова. В данном случае статья и пример на английском языке;

Все они содержатся в единой структуре T∗ = {n ∗ , v∗ , s∗ , p∗} (визуально проще воспринимать в виде таблицы, как на рис. 1).

На этапе планирования создается структура предложения, основанная на выбранных словах. Это включает в себя определение грамматических ролей слов и их последовательности в предложении, что критично для естественного звучания текста.

На этапе реализации создаётся структурированный естественный для восприятия человеком текст. Для этого используются различные способы, например:

- **Шаблонные методы:** Используют заранее заданные структуры предложений, такие как "Объект + Действие + Место", что позволяет генерировать текст на основании шаблонов;

- **Грамматические модели:** Алгоритмы, которые применяют правила грамматики для построения синтаксически верных предложений;

**- Статистические модели:** Развиваются на основе вероятностных методов;

- **N-Граммовые модели:** Определяют вероятность появления слова на основе N предыдущих слов. Эти модели широко использовались для обработки текстов до появления глубокого обучения;

- **Hidden Markov Models:** Модели для обработки последовательных данных, используются для генерации текстов из скрытых элементов, таких как части речи;

- **Модели на основе машинного обучения**: Эти модели начали доминировать в последние годы. Они требуют больших затрат на создание и поддержание, но показывают достаточно высокую эффективность относительно многих моделей;

- **LSTM (Long Short-Term Memory):** Рекуррентные нейронные сети, которые обеспечивают сохранение контекста в длительных последовательностях при генерации текста;

- **GRU (Gated Recurrent Unit):** Подобные LSTM модели, но с меньшим количеством параметров, которые также эффективны для генерации текста;

- **Глубокие нейронные сети и трансформеры:** Современные и наиболее продвинутые подходы;

- **Трансформеры:** Используют механизм внимания для обработки последовательностей информации, что позволяет моделям эффективно учитывать контекст. Примеры включают модели, такие как BERT и GPT (Generative Pre-trained Transformer), которые демонстрируют выдающиеся результаты в генерации естественного языка;

- **Генеративные модели:** Модели, которые фокусируются на создании нового контента;

- **GANs (Generative Adversarial Networks):** Характеризуются двумя компонентами — генератор и дискриминатор; применяются для создания правдоподобных текстов или изображений;

## **1.5 Где может использоваться машинное зрение и генерация текста?**

Вместе или раздельно данные технологии могут использоваться в различных системах. Такие есть как зарубежные, так и российские, например:

- **Google Cloud Vision API:** компания Google предоставляет возможность пользоваться технологией image captioning через свои серверы с помощью API. Данная услуга широко распространена и ей пользуется большое количество компаний в своих продуктах для автоматической разметки изображений, повышения эффективности функций поиска и других задач;

- **Социальные сети:** VK, а также запрещённые на территории Российской Федерации Facebook, Instagram используют технологии image captioning для генерации описаний к фото, определения пользователей на фотографиях. Pinterest использует технологию для создания описаний к изображению при сохранении их на свои «доски»;

- **Медицина:** аппараты МРТ или устройства анализа рентгеновских снимков используют технологию image captioning для генерации описания потенциальных заболеваний;

- **AI в роботах:** такие компании, как Tesla или Yandex разрабатывают системы анализа окружения в реальном времени. Для этого используются и совершенствуются методы и алгоритмы технологии image captioning;

Данная технология может использоваться не только в узконаправленных сферах науки, но и в сфере развлечения. Подобные технологии предоставляет в качестве услуг компания OpenAI. Чат-бот Chat GPT умеет распознавать изображения и давать текстовое описание. Реализован бот на сайте. Данный функционал возможно реализовать и в приложении;

# **ГЛАВА 2**

## **2.1 В ходе анализа данных по теме image captioning выяснилось:**

Для создания приложения с функционалом этой технологии необходимо извлекать ключевые данные из изображения (как на рис. 1), затем преобразовывать в текст.

Для такого наиболее удачным вариантом будет язык программирования Python. Внутри обширное количество библиотек, где уже реализованы данные возможности.

## **2.2 Для этого будут использованы ключевые библиотеки:**

- **Pillow** для обработки изображений;

Pillow и его предшественник PIL — это оригинальные библиотеки Python для работы с изображениями. Несмотря на то, что существуют другие библиотеки Python для обработки изображений, Pillow остается важным инструментом для понимания и работы в целом. Для оперирования и обработки изображений Pillow предоставляет инструменты, аналогичные тем, которые можно найти в программном обеспечении, таком как Photoshop. Некоторые из более современных библиотек обработки изображений Python построены на основе Pillow и часто предоставляют более продвинутую функциональность.

- **ImageAI** для компьютерного зрения;

ImageAI — это простая, но очень мощная и продвинутая библиотека Python с открытым исходным кодом, которая дает разработчикам программного обеспечения возможность разрабатывать приложения и программные утилиты с автономными возможностями глубокого обучения и компьютерного зрения.

- **Transformers** для получения текста на естественном языке;

Transformers — это мощная библиотека Python, созданная Hugging Face, которая позволяет вам загружать, управлять и запускать тысячи предварительно обученных моделей ИИ с открытым исходным кодом.

- **Tkinter** для создания графического интерфейса приложения;

Tkinter – это пакет для Python, предназначенный для работы с библиотекой Tk. Библиотека Tk содержит компоненты графического интерфейса пользователя (graphical user interface – GUI). Эта библиотека написана на языке программирования Tcl.

## **2.3 Для реализации работы функционального GUI необходимо:**

- Поле или кнопка для загрузки изображений: это основной элемент интерфейса, который позволяет пользователю загружать изображения в приложение. Он может быть представлен в виде кнопки, по нажатию на которую открывается диалоговое окно для выбора файла, или в виде текстового поля, куда пользователь может перетаскивать изображения. Важно, чтобы этот элемент был интуитивно понятным и доступным, чтобы пользователи могли легко загружать изображения без лишних усилий.

- Поле для вывода строки текста: это текстовое поле предназначено для отображения результатов обработки изображений. После того как изображение будет загружено и обработано, в этом поле будет выводиться сгенерированный текст, который описывает содержимое изображения. Поле должно быть достаточно большим, чтобы отображать длинные строки текста, и иметь возможность прокрутки, если текст превышает видимую область.

- Возможность использовать функцию «Обзор»: эта функция значительно упрощает процесс загрузки изображений. При нажатии на кнопку «Обзор» открывается стандартное диалоговое окно, где пользователь может выбрать одно или несколько изображений из файловой системы. Это позволяет избежать необходимости вручную вводить пути к файлам и делает процесс более удобным и быстрым.

## **2.4 Общий функционал должен включать:**

Общее функциональное наполнение приложения должно включать в себя несколько ключевых возможностей, которые обеспечат его эффективность и полезность для пользователей:

- **Загрузка и вывод изображения:** приложение должно поддерживать загрузку изображений в различных форматах (например, JPEG, PNG). После загрузки изображение должно отображаться в интерфейсе, чтобы пользователь мог видеть, что именно он загрузил. Это важно для подтверждения правильности выбора изображения.

- **Обработка изображения для получения объектов с ключевыми словами:** после загрузки изображения приложение должно автоматически обрабатывать его с использованием алгоритмов компьютерного зрения. Это включает в себя идентификацию объектов на изображении и извлечение ключевых слов, которые описывают эти объекты. Например, если на изображении изображены собака и мяч, приложение должно распознать эти объекты и сохранить их в виде ключевых слов. Для обнаружения будет использована модель YOLO.

- **Генерация текста на естественном языке:** на основе полученных данных с объектов приложение должно генерировать текст на естественном языке. Это может быть описание изображения, составленное на основе распознанных объектов. Например, для изображения с собакой и мячом приложение может сгенерировать текст: «На изображении собака играет с мячом». Генерация текста должна быть выполнена с использованием современных моделей обработки естественного языка, чтобы обеспечить высокое качество и естественность текста.

- **Вывод текста в GUI пользователю:** сгенерированный текст должен быть отображен в текстовом поле, упомянутом ранее. Это позволит пользователю быстро ознакомиться с результатами обработки изображения. Важно, чтобы текст был четким и легко читаемым, а также чтобы пользователь мог копировать его при необходимости.

- **Возможность загрузки сразу нескольких изображений:** для повышения удобства работы с приложением пользователи должны иметь возможность загружать несколько изображений одновременно. Это может быть реализовано через диалоговое окно выбора файлов, где пользователь может выделить несколько изображений. Приложение должно обрабатывать каждое изображение по очереди и генерировать соответствующий текст для каждого из них, что значительно упростит процесс внесения данных и повысит эффективность работы.

Эти функциональные требования обеспечивают создание интуитивно понятного и эффективного графического интерфейса, который позволит пользователям легко загружать изображения, обрабатывать их и получать текстовые описания. Уделяя внимание каждому из этих аспектов, можно создать приложение, которое будет полезным и удобным для пользователей, а также соответствовать современным стандартам разработки программного обеспечения.

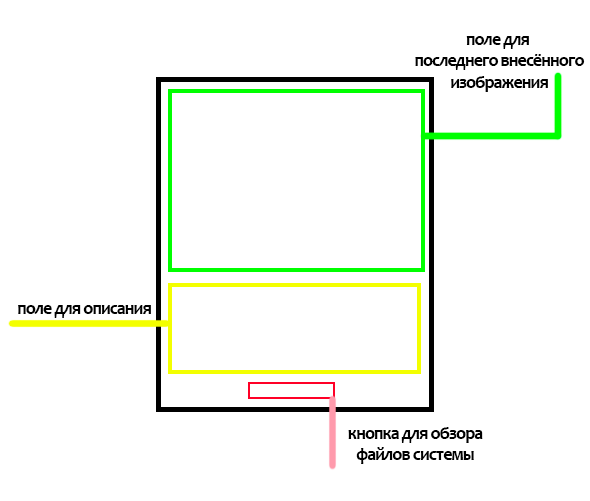
****

Рис. 2 Примерное изображение GUI

### **ГЛАВА 3**

Доделать генерацию естественного текста

Поискать новые версии модели YOLO (если не будет, оставить имеющуюся на данный момент. Спросить нужно ли обучать модель)

Сделать вывод сэмплов в отдельную папку

Поменять инструменты в заявлении

https://github.com/ElScorpionwin/MIREA\_COURSE\_PAPER

*1 глава анализ теории,решений, требования к своей работе*

*2 глава: концепция*

*Выбор инструментов для создания*

*Дизайн интерфейса*

*3 глава:*

*Пояснение основных кусков кода*

*Инструкция как пользоваться продуктом*

*В заключении цель достигнута или нет и с каким результатом решены задачи*

*Список литературы ссылка, название, автор/коллектив авторов, дата на день просмотра*