

Автоматизированная разметка изображений с беспилотного летательного аппарата при помощи предобученных моделей искусственного интеллекта

Н.А. Бессарабов^{1✉}, А.А. Сапожников¹, Д.В. Татарников¹,
Р.Р. Тюгунов¹, А.М. Цыганов¹

¹ АНО «КРР «Мой район», г. Москва, 125009, Россия

Ссылка для цитирования

Бессарабов Н.А., Сапожников А.А., Татарников Д.В., Тюгунов Р.Р., Цыганов А.М. Автоматизированная разметка изображений с беспилотного летательного аппарата при помощи предобученных моделей искусственного интеллекта // Программные продукты и системы. 2025. Т. 38. № 2. С. 251–260. doi: 10.15827/0236-235X.150.251-260

Информация о статье

Группа специальностей ВАК: 1.2.1

Поступила в редакцию: 07.06.2024

После доработки: 28.08.2024

Принята к публикации: 09.09.2024

Аннотация. В работе рассматривается метод разметки изображений, полученных с беспилотного летательного аппарата, при помощи предобученных моделей искусственного интеллекта для последующего обучения детекторов объектов различных классов. Подход состоит из трех основных этапов: сегментация изображения, выделение зон внимания, проверка сегментов на наличие целевого класса при помощи комитета большинства из мультимодальных моделей. Для сегментации используется модель Segment Anything Model, предназначенная для объектной сегментации. Так как сегментов может быть значительное количество, предложен алгоритм построения иерархической структуры над множеством сегментов, то есть если какой-либо сегмент содержится в другом, то этот сегмент будет вложен в содержащий его сегмент. При дальнейшей обработке для значительного ускорения применяется следующая логика: если в родительском сегменте целевой класс не обнаружен, то и во вложенных сегментах его точно не будет. На втором этапе предложен алгоритм нахождения зон внимания как наиболее вероятных зон нахождения объектов целевого класса. Для этого используется предобученная модель SigLIP. С помощью зон внимания все сегменты из обрабатываемого множества изображений ранжируются по убыванию вероятности нахождения в них целевого класса. На третьем этапе применяется комитет большинства из трех мультимодальных чат-ботов – LLaVA, CogVLM, Mini-Gemini. Для проверки чат-ботами предусмотрено правило остановки, чтобы избежать полного перебора всех сегментов и тем самым ускорить процесс разметки изображений. Проведен вычислительный эксперимент для демонстрации метрик и скорости работы предложенного подхода.

Ключевые слова: датасет, сегментация изображений, большие языковые модели, чат-бот, нейронные сети, генеративный ИИ, разметка

Введение. Дронопорт представляет собой программно-аппаратный комплекс, выполняющий функцию базовой станции для *беспилотных летательных аппаратов* (БПЛА), с которой они взлетают для решения задач по съемке территории. Он позволяет обеспечивать круглосуточный мониторинг заданной территории. В связи с этим возникают практические задачи компьютерного зрения, в частности, детектирования различных объектов на изображениях или в видеопотоке, например, обнаружение людей, выявление неправильной парковки, свалок, незаконных построек, детекция тропинок и др.

Для обработки большого количества данных и обнаружения различных объектов требуется обучать соответствующие детекторы. Каждый детектор должен работать достаточно быстро, точно и находить почти все объекты. Классический подход к решению задачи осложняется тем, что для получения высокой

точности и полного охвата интересующих объектов требуется большая выборка и, как следствие, на разработку такого детектора необходимо много времени. Разметка обучающей выборки может быть ускорена за счет расширения штата разметчиков, однако даже в этом случае остается острым вопрос качества подготовленной разметки.

В данной работе исследуется оригинальный подход к подготовке выборки для последующего обучения детектора. В его рамках для разметки данных применяются несколько предобученных моделей (подход к детекции объектов zero-shot), в том числе мультимодальные большие языковые модели и специально созданные алгоритмы постобработки результатов сегментации, построения зон внимания, правила остановки и обхода сегментов чат-ботами. Данный подход позволяет достичь высокой полноты поиска целевых объектов, ускорить разметку и повысить ее качество.

Постановка задачи

Цель работы состоит в создании системы автоматизированной разметки изображений для детектирования объектов.

Разметка заключается в том, что на каждом обрабатываемом изображении искомый объект выделяется контуром и задается его метка. Для оценки качества работы системы небольшая часть набора данных размечается вручную и считаются метрики *precision* (точность) и *recall* (полнота). Поскольку часть набора данных размечаются вручную, предполагается, что быстрее разметить целевой класс с помощью ограничивающих рамок. Для определения всех перечисленных метрик сначала находится коэффициент *IoU* (*Intersection over Union*), равный отношению площади пересечения ограничивающих рамок предсказания и истинной разметки к их объединению. Для метрик точность и полнота выбирается пороговое значение *IoU*, при котором предсказание считается совпадающим с истинной разметкой. Такой порог задается индивидуально для каждого целевого класса.

Кроме того, важным параметром работы системы разметки, напрямую влияющим на эффективность практического использования, является скорость работы. Быстродействие системы разметки зависит от многих факторов, в том числе от самих изображений, специфики искомого объекта, от алгоритма обхода объектов на изображении, скорости работы предобученных моделей, от производительности вычислительной платформы.

Таким образом, задача состоит в том, чтобы за разумное время найти на пачке изображений контуры интересующих объектов при изначально заданных порогах *IoU* для каждого класса с метриками *precision* и *recall*, не хуже заранее заданных.

В данной работе не рассматриваются вопросы, касающиеся полетных миссий и условий съемки, в частности, влияние погодных условий и времени суток. Предполагается, что на вход для разметки поступит набор изображений, сделанных одинаковыми камерами примерно с равной высоты и при схожем угле наклона камеры. Кроме того, на всех изображениях достаточно освещенности, чтобы контуры искомого объекта были четко различимы.

Практическая значимость работы состоит в создании системы разметки с высокими показателями полноты и точности разметки,

а также в ускорении разметки за счет обработки не всех сегментов, а наиболее вероятных. Научная новизна – в использовании для проверки сегмента не одной модели, а комитета большинства из нескольких моделей, а также в создании алгоритмов выделения зон внимания и правила остановки при переборе сегментов.

Анализ литературы

Вопросы компьютерного зрения возникают во многих практических задачах. Значительные успехи в этой области были связаны с развитием глубоких нейронных сетей. Так, в работе [1] предложена реализация методов обучения глубоких нейронных сетей, оптимизированная для расчета на графическом процессоре, что позволило повысить производительность сети и ускорить время ее обучения. Однако переход от методов классического машинного обучения к глубокому обучению сопряжен с двумя новыми сложными проблемами: во-первых, с низкой скоростью обучения нейронных сетей с нуля, во-вторых, с необходимостью сбора больших наборов данных для обучения.

Следующим этапом развития стала парадигма с использованием предобученных моделей с последующей тонкой настройкой [2, 3]. В этом случае нейронная сеть сначала предварительно обучается на больших датасетах, а затем уже обученная модель подвергается тонкой настройке с помощью аннотированных обучающих данных для конкретных задач. Благодаря обширным знаниям, полученным в предварительно обученных моделях, такая парадигма обучения может ускорить обучение нейронной сети под конкретную задачу и обеспечить высокое качество работы модели. Схема показала большую эффективность в широком спектре задач компьютерного зрения, тем не менее она по-прежнему требует дополнительного этапа тонкой настройки для конкретной задачи с размеченными данными обучения для каждой последующей задачи.

Учитывая значительные достижения в области обработки естественного языка [4], в частности, успешное дообучение предобученных моделей семейства *Bert* для различных задач и совершенствование генеративных текстовых моделей, получил развитие новый подход. Он заключается в предварительном обучении визуально-языковой модели (мульти-модальной, *Vision Language Model*, *VLM*) и в последующем применении модели без тонкой настройки (подход *zero-shot*) [5]. В рам-

как этого подхода мультимодальная модель предварительно обучается на огромных датасетах в виде изображение-текст, которые достаточно легко собираются в Интернете. С учетом этого подхода активно развиваются несколько направлений: во-первых, адаптация мультимодальных моделей под различные задачи, в частности, подготовка текстового запроса к модели для повышения качества ее работы [6, 7], борьба с галлюцинациями, во-вторых, дистилляция моделей с целью повышения их производительности.

Для использования в дальнейших исследованиях были отобраны три мультимодальные модели: LLaVa 1.6 (*Large Language and Vision Assistant*), CogVLM (*Cognitive Vision-Language Model*), Mini-Gemini. Каждая из них имеет высокие метрики в задаче ответа на вопрос по изображению VQA (*Visual Question Answering*). Каждый из рассматриваемых чат-ботов может иметь в качестве основы различные текстовые и визуальные энкодеры, в связи с чем представляется целесообразным оценить метрики и каждой модели, и ансамбля из трех моделей.

С учетом того, что на изображении может быть сразу несколько различных объектов и каждый из них может быть интересующим объектом целевого класса, для сведения рассматриваемой в исследовании задачи к задаче VQA необходимо выделять на изображении отдельные объекты. В данном исследовании выделение отдельных объектов выполняется с помощью сегментации изображения [8]. Чаще всего выделяют два вида сегментации – семантическую и объектную. Семантическая сегментация – это процесс присвоения метки каждому пикселю изображения. Она рассматривает несколько объектов одного класса как одну сущность. При объектной сегментации объекты разделяются на отдельные экземпляры, даже если относятся к одному классу. Пример: определение беспилотным автомобилем других машин, чтобы правильно их классифицировать и выявлять точные контуры каждого для безопасного движения. Модели объектной сегментации не знают, к какому классу принадлежит классифицируемая область, но могут разделить перекрывающиеся или очень похожие области объекта на основе их границ. Обычно сегментация отдельных объектов сложнее, чем семантическая сегментация, так как при сегментации отдельных объектов нужно не только классифицировать каждый пиксель, но и разделить объекты одного класса на отдельные экземпляры, и при этом не только определить класс

объекта, но и точно выделить границы каждого объекта, даже если они принадлежат одному классу. Например, если на изображении несколько человек, то каждому нужно присвоить уникальную метку. При семантической сегментации каждому пикселю изображения нужно присвоить метку класса. Необходимо различать классы объектов, но не требуется разделять объекты одного класса. Например, если на изображении несколько людей, все они будут отмечены как «человек» без разделения на отдельные экземпляры. В соответствии со спецификой задачи, учитывая существенное разнообразие распознаваемых объектов, представляется целесообразным применять модель объектной сегментации изображений SAM (*Segment Anything Model*) [9].

Одной из особенностей работы модели сегментации является создание сетки над изображением и поиском контуров исходя из пересечения с этой сеткой. Если искомый контур попадает внутрь сетки, то он не будет найден. Следовательно, для модели сегментации необходимо задать размер сетки таким образом, чтобы гарантированно находились целевые объекты минимального размера. Помимо этого, при больших изображениях модель сегментации может требовать для работы слишком много видеопамяти, в связи с этим большие изображения приходится сжимать, а затем соответствующим образом сжимать и сетку.

Архитектура системы авторазметки

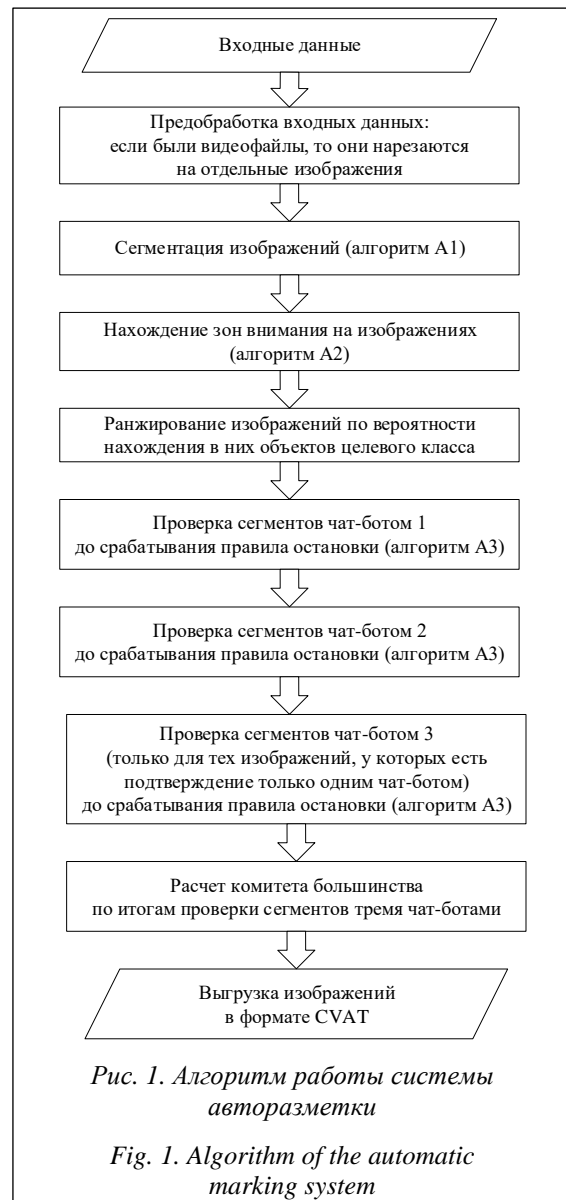
Предполагается, что перед началом работы задан целевой объект: потенциальные размеры, визуальные характеристики, различные синонимы, которыми он может обозначаться. Работа системы разметки данных состоит из нескольких основных этапов. На первом этапе каждое изображение сегментируется, то есть в итоге для каждого изображения на выходе получается набор контуров. Необходимо проверить, есть ли на каждом из этих контуров целевой объект. Так как различных сегментов немало, а многие из них либо пересекаются, либо входят друг в друга, предложен алгоритм создания иерархии сегментов (алгоритм A1). На втором этапе выполняется алгоритм нахождения зон внимания на изображениях (алгоритм A2), в результате работы которого каждое изображение разбивается на небольшие прямоугольники с вычисленной вероятностью нахождения в них искомого объекта. Исходя из этой вероятности каждому сегменту присваивается

вероятность нахождения в нем целевого объекта. На третьем этапе каждый сегмент проверяется чат-ботом. Для повышения качества в системе применяется алгоритм А3 и используются сразу несколько различных чат-ботов, а итоговое решение принимается на основе комитета большинства. Проверка чат-ботами идет по ранжированной таблице сегментов. С учетом того, что проверять все сегменты наверняка нет смысла, задается условие остановки: если на N сегментах подряд объект целевого класса не найден, то ни один последующий сегмент на проверку чат-ботом не попадает. Далее подготовленная разметка может быть выгружена в систему аннотирования, в частности, в CVAT (*Computer Vision Annotation Tool*), где разметчики могут ее проверить и при необходимости изменить (<https://github.com/cvat-ai/cvat>). Алгоритм работы системы приведен на рисунке 1.

Стоит отметить, что алгоритмы А1 и А2 могут выполняться либо последовательно, либо одновременно в зависимости от возможностей вычислительного кластера. Кроме того, подобный алгоритм хорошо подходит для задач разметки сразу нескольких целевых классов, когда высока вероятность найти какие-либо объекты целевого класса на каждом из изображений (например, на одном изображении найдутся примеры неверной парковки автомобиля, а на другом – мусор), то есть это случаи, когда необходимо сегментировать все изображения. Если нужно отыскать лишь один объект целевого класса, целесообразно применять вначале алгоритм А2, затем отбирать изображения, где наиболее высока вероятность найти целевые объекты, а сегментировать уже лишь их и, соответственно, проверять сегменты только для них.

Подробнее этапы описаны далее. Кроме того, для каждого из них рассмотрен вопрос скорости работы (она будет оценена на вычислительном кластере с характеристиками: 32vCPU 2,6GHz, 160 GB RAM, 1vGPU A100 80GB). Для оценки метрик работы системы используется датасет [10], в котором содержатся 420 изображений городских пейзажей, снятых с дрона камерой с разрешением 4К (то есть 3 840×2 160 пикселей) под углом примерно 45 градусов к горизонту на высоте от 50 до 100 метров. Изображения в датасете размечены сегментами на несколько классов, в том числе здания, дороги, деревья, невысокая растительность, автомобили, люди.

Для разработки системы использован язык программирования Python 3.12. Для каждой



применяемой модели приведены ссылки на исходный код. Если исходная модель доступна на портале huggingface.co, то для единообразия кода используется модель с этого портала.

Сегментация изображений

Для сегментации изображений выбрана модель SAM [9]. Стоит отметить, что модель сегментации может требовать значительных временных затрат, поэтому минимальный размер целевого объекта должен быть подобран исходя из допустимых временных затрат на каждое изображение. На тестовом наборе данных для сегментации одного изображения при поиске объекта с минимальным размером 30 пикселей требовалось в среднем 29 секунд.

Сегментатор для каждого изображения возвращает достаточно много сегментов, причем часть из них могут пересекаться между собой или даже полностью входить друг в друга. Поскольку система должна работать быстро, представляется целесообразным распределять все сегменты в иерархическую структуру, что выполняется применением алгоритма A1.

Алгоритм A1 – сбор иерархии сегментов (постобработка результатов сегментации).

1. Нахождение сегментов, которые полностью (с некоторым допуском, в работе использовался допуск 90 %) входят в другой сегмент. Каждая пара сегментов выстраивается в иерархию, где внутренний сегмент является дочерним, а внешний – родительским. Этап выполняется до тех пор, пока не останется ни одной пары сегментов, входящих друг в друга. Для этого этапа каждый сегмент переводится в полигон и обрабатывается с помощью библиотек *shapely* и *geopandas*.

2. Нахождение пересекающихся сегментов (пересекаются более некоторого допуска, который менее допуска на полное вхождение; в работе использовался допуск 20 %). Для каждой пары пересекающихся сегментов создается сегмент, равный объединению пересекающихся сегментов, который является для них родительским, а оба исходных сегмента вложены в него. Этап выполняется несколько раз до тех пор, пока не останется ни одной пары сегментов, входящих друг в друга.

3. Для каждой пары сегментов проверяется пересечение их выпуклых оболочек. Если пересечение есть, то, как и на предыдущем этапе, создается сегмент, равный пересечению выпук-

лых оболочек двух сегментов, который является родительским для них, исходные сегменты вложены в него. Этот этап хорошо зарекомендовал себя, например, при поиске мусора, когда сегментатор выделял отдельные элементы (доски, крышки и т.д.), а на этом этапе алгоритма создавался сегмент с кучей мусора.

4. Добавление остаточных сегментов – части изображений, которые не вошли ни в один сегмент. Остаточный сегмент зачастую представляет собой один большой сегмент, который состоит из нескольких крупных частей, объединенных тропинками. На данном этапе задается буфер (в работе использован буфер 15 пикселей), меньший минимального размера искомого объекта, а затем этот остаточный сегмент сжимается по всем границам на буфер, что позволяет разбить большой сегмент на несколько небольших.

Алгоритм дает возможность построить иерархическую структуру над полученными сегментами. Это будет использовано для ускорения проверки сегментов чат-ботами исходя из следующей логики: если целевой объект не найден в родительском сегменте, то и во всех вложенных сегментах целевых объектов не будет. На тестовом наборе данных на каждом изображении в среднем возвращалось порядка 400 сегментов, а за счет иерархии на первом уровне удавалось получать в среднем 200 сегментов.

На рисунке 2 представлен пример сегментации изображения: пунктирные черные линии обозначают границы исходных сегментов, желтые области – объединенные сегменты, фиолетовые – выпуклые оболочки над группами сег-

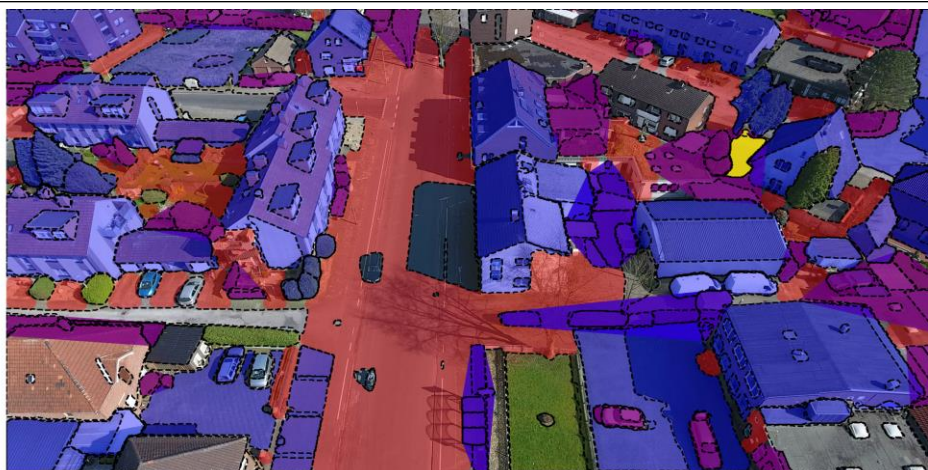


Рис. 2. Пример работы сегментатора

Fig. 2. Example of segmenter operation

ментов, синие – вложенные сегменты, а красные – остатки.

Выделение зон внимания на изображении

Для нахождения зон внимания на изображении выбрана модель SigLIP (*Sigmoid loss for Language-Image Pre-training*) [11]. Она принимает на вход изображение или набор изображений, а также набор ключевых слов, описывающих целевой класс. Выбранная модель обрабатывает входное изображение, которое сжимается или растягивается до размера 384×384 пиксела, и возвращает вероятность нахождения на таком изображении объекта, описанного словосочетанием. Для нахождения зон внимания использован алгоритм A2.

Алгоритм A2 – построение зон внимания.

1. Задаются искомые классы и одно или несколько словосочетаний для каждого из них.

2. Задаются одно или несколько разбиений изображений на квадраты. Для обработки данного датасета использованы четыре разбиения: первое начиналось с точки (0;0), второе – с (0;192), третье – с (192;0), четвертое – с (192; 192). Размер квадрата в каждой сетке – 384×384 пиксела.

3. Для каждого квадрата из каждого разбиения с помощью модели SigLIP находится вероятность нахождения объектов каждого искомого класса.

4. Для каждого из квадратов размера 192×192 пиксела сначала находится среднее значение

вероятности для всех словосочетаний и всех классов среди четырех используемых разбиений, а затем максимальное значение вероятности для каждого из искомых классов, что и является результатом работы алгоритма.

Пример работы алгоритма для поиска автомобилей показан на рисунке 3. Без сдвига на одном изображении с разрешением 4К (то есть 3840×2160 пикселей) модель работает 4 секунды, со сдвигами на одно значение (4 сетки) примерно 15 секунд. Время работы прямо пропорционально квадрату числа сдвигов в сетке.

После того как сегментировано изображение и рассчитаны зоны внимания для изображения, все сегменты пересекаются с зонами внимания. Для каждого сегмента вычисляется значение вероятности как средневзвешенное значение вероятности в прямоугольнике с весом – доля площади квадрата в сегменте. Результат работы представлен на рисунке 4.

Применение мультимодальных больших языковых моделей для анализа сегментов

После предыдущего этапа готова плоская таблица со всеми сегментами для всех изображений, причем для каждого сегмента определены, во-первых, вероятность нахождения в нем объекта целевого класса, а во-вторых, уровень иерархии этого сегмента. Для итоговой проверки наличия объекта целевого класса в каждом сегменте представляется целесообразным использовать комитет большинства

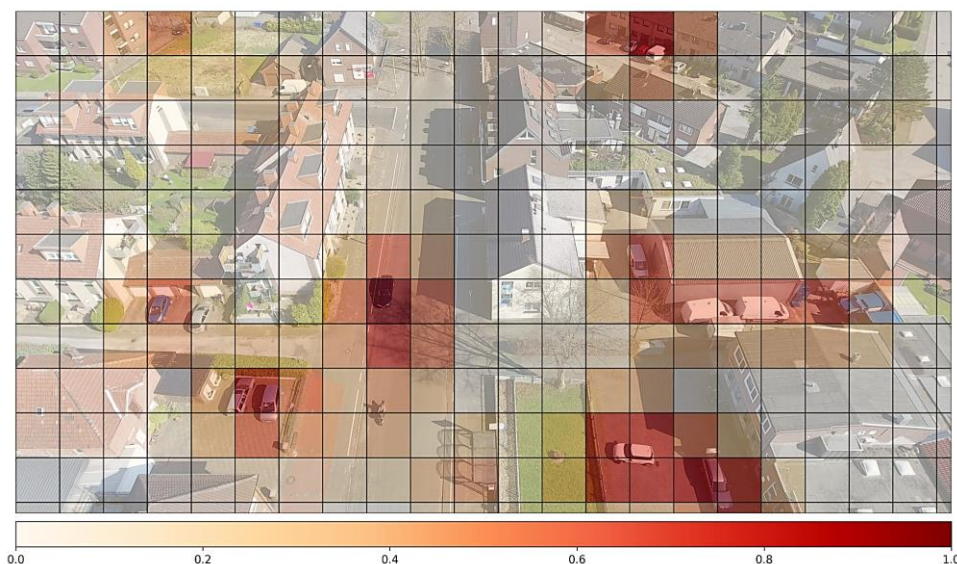
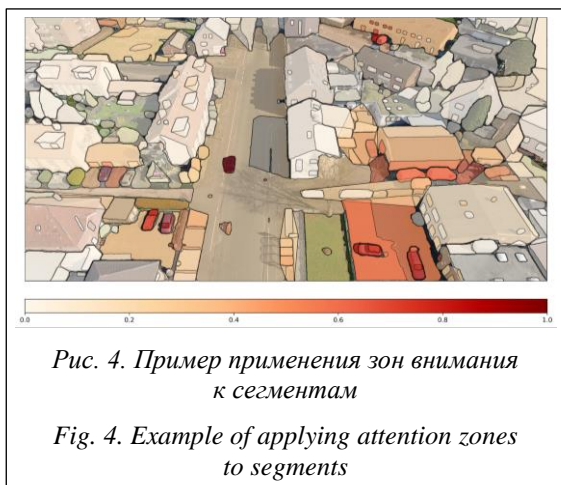


Рис. 3. Пример нахождения зон внимания на изображении

Fig. 3. Example of finding attention zones in an image



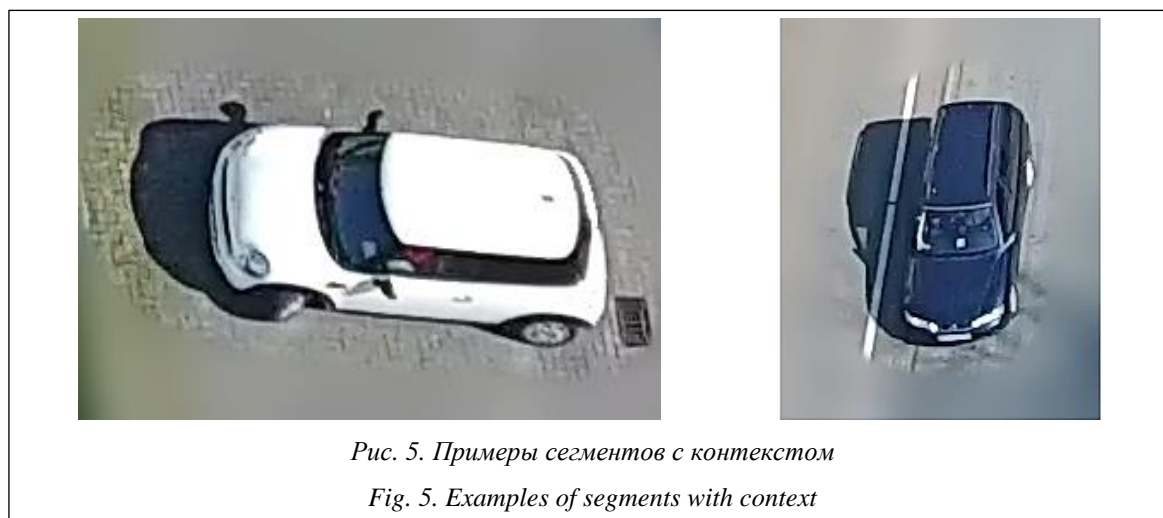
из нескольких моделей. В частности, в данной работе выбраны три предобученных чат-бота: LLaVa 1.6 (llava-v1.6-mistral-7b-hf), CogVLM (cogvlm-chat-hf), Mini-Gemini (MGM-13B-HD). Для каждой из этих моделей рассмотрены варианты запуска как с одним промптом, так и с несколькими на случай поиска сразу нескольких целевых классов, а также работа сразу с несколькими изображениями в рамках одного запроса. LLaVa 1.6 может работать сразу и с несколькими запросами, и с несколькими изображениями в одном запросе; при проверке одного целевого класса удавалось запустить чат-бот с 40 изображениями в одном запросе, благодаря чему средняя скорость обработки одного изображения составила около 175 мс. Чат-бот CogVLM не в состоянии работать с несколькими классами сразу, но может работать с несколькими изображениями в одном запросе; при проверке одного целевого класса удавалось запустить чат-бот с 30 изображениями в одном запросе, благодаря чему средняя скорость обработки одного изображе-

ния составила около 205 мс. Mini-Gemini может работать сразу на проверку нескольких классов, но не работает с несколькими изображениями в одном запросе; при проверке одного класса один сегмент проверяется в среднем за 430 мс. Исходя из этого выбран следующий порядок применения моделей: 1 – LLaVa 1.6, 2 – CogVLM, 3 – Mini-Gemini.

Для каждого чат-бота необходимо подобрать выражение-вопрос, на который виртуальный ассистент будет давать ответ. Представляется целесообразным создать небольшую выборку изображений с предварительно размеченными целевыми сегментами и предложить каждому из чат-ботов их описать. Тогда для каждого чат-бота будут получены наборы слов и словосочетаний для обозначения объектов, которые мультимодальный чат-бот видит на изображении. Исходя из этого можно собрать целые предложения, например, для всех чат-ботов и класса «автомобиль» был подобран следующий запрос: Answer just yes or no in one word. Are there cars, automobiles and vehicles that are clearly visible in the image?

Каждый сегмент подается в чат-бот вместе со своим контекстом, так как для модели может быть важно окружение объекта, а помимо этого, резкий край может привести к галлюцинациям модели. В связи с этим для каждого сегмента сначала вырезается область из исходного изображения с отступом от сегмента, равным 10 % линейного размера сегмента, в каждую из сторон. Затем вырезанное изображение зашумляется, а сверху на зашумленное изображение накладывается сам сегмент. Примеры сегментов с контекстом показаны на рисунке 5.

Обход сегментов чат-ботами осуществляется в соответствии с алгоритмом А3.



Алгоритм А3 – обход сегментов чат-ботами.

1. Все сегменты ранжируются по вероятности нахождения в них искомого объекта по убыванию.

2. Проверка каждого сегмента самой быстрой моделью LLaVa 1.6. Модель проверяет все сегменты в порядке убывания вероятности до тех пор, пока не будет *N* подряд идущих нулей, то есть сегментов, на которых искомым класс не найден. Кроме того, после обработки каждого батча проверяется наличие искомого класса: если для какого-то родительского сегмента он не найден, то и для всех вложенных сегментов проставляется 0, то есть результат не найден. Для проверки отбираются только те сегменты, у которых еще нет результата проверки.

3. Аналогично п. 2 происходит обработка сегментов второй моделью (CogVLM).

4. Для проверки третьей моделью (Mini-Gemini) отбираются только те сегменты, у которых лишь одна из двух моделей нашла искомым объект. Этот чат-бот проверяет все отобранные сегменты без правила остановки. На тестовом наборе для третьей модели оставалось не более 10 % от всех исходных сегментов.

5. На основе результатов проверки сегментов тремя моделями рассчитывается комитет большинства: если две и более моделей нашли искомым объект в сегменте, то результат «найден», иначе – «не найдено»

Таким образом, будет подготовлена таблица с набором контуров, в которых комитет большинства из трех чат-ботов подтвердил наличие объектов целевого класса. Для проверки работы эти контуры выгружаются в систему CVAT, где разметчик может проверить подготовленную разметку данных, при необходимости удалить или доработать ее.

Экспериментальное исследование системы авторазметки

Для тестирования системы авторазметки были выбраны 420 изображений автомобилей (объединение классов static car и moving car). Из них 20 использовались для настройки ключевых слов и промптов SigLip, LLaVa 1.6, CogVLM, Mini-Gemini, а также параметров сегментатора. На остальных 400 изображениях, которые также были размечены, рассчитывались метрики работы системы.

В рамках эксперимента сегментация каждого изображения занимала в среднем 29 секунд, расчет зон внимания – 15 секунд, работа чат-бота LLaVa 1.6 – 175 мс на сегмент, CogVLM – 205 мс на сегмент, Mini-Gemini – 430 мс на сегмент.

В таблице приведены метрики precision и recall, а также суммарное время работы для каждого чат-бота и комитета большинства. Для оценки влияния зон внимания и правила остановки на скорость и качество работы были даны два варианта метрик: с использованием правила остановки (1-е значение) и без его использования (2-е значение).

Результаты, представленные в таблице, свидетельствуют о близких значениях итоговых метрик обработки данных с учетом и без учета зон внимания и правил остановки, но существенном, более чем в три раза, ускорении процесса разметки. При этом комитет большинства обладает в совокупности лучшим соотношением точности, полноты и F-меры по сравнению с каждой из моделей.

Общее время работы системы с учетом сегментации изображений, расчета зон внимания и поиска целевых объектов с помощью чат-ботов составило 1 037 минут, или около 2,5 минуты на одно изображение. При этом на всех изображениях присутствовало 16 539 сегментов с ма-

Метрики работы каждой модели и системы в целом

Performance metrics for each model and the system in general

Модель	Recall		Precision		F-мера		Количество проверенных сегментов		Общее время работы, мин.	
	1	2	1	2	1	2	1	2	1	2
LLaVa 1.6	0,8376	0,8363	0,6620	0,6414	0,7395	0,7260	96 375	177 542	281	518
CogVLM	0,9086	0,9081	0,5388	0,5193	0,6765	0,6607	110 312	177 542	377	607
Mini-Gemini	0,5130	0,8121	0,5391	0,7158	0,5257	0,7609	9 942	177 542	71	1 272
Комитет большинства из 3 моделей	0,8734	0,8742	0,6599	0,6629	0,7518	0,7540			729	2 397

шинами, найдено было 14 445 из них, а помимо этого, еще 7 446 ошибочных сегментов, которые в итоге могут проверить разметчики.

Заключение

Разработана система автоматизированной разметки для обучения детекторов в задачах детекции объектов. Проведены эксперименты для оценки качества работы системы, а также времени ее работы. Предложенные алгоритмы постобработки результатов сегментации, построения зон внимания, правила остановки и обхода сегментов чат-ботами обладают высокой полнотой поиска целевых объектов и вы-

сокой точностью, а также позволяют ускорить процесс поиска по сравнению с полным перебором более чем в три раза.

Данный подход может найти применение при дальнейшем развитии детектора: когда первичный детектор уже обучен, он может быть задействован для разметки данных, дополняя или проверяя разметку, подготовленную с использованием созданных алгоритмов.

Практическая значимость исследования состоит в том, что разработанные алгоритмы и методики позволяют достичь высокой полноты поиска целевых объектов и точности, а также ускорить процесс обработки информации.

Список литературы

1. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 2017, vol. 60, no. 6, pp. 84–90. doi: 10.1145/3065386.
2. He K., Fan H., Wu Y., Xie S., Girshick R. Momentum contrast for unsupervised visual representation learning. *Proc. IEEE Conf. CVPR*, 2020, pp. 9726–9735. doi: 10.1109/CVPR42600.2020.00975.
3. Chen T., Kornblith S., Norouzi M., Hinton G. A simple framework for contrastive learning of visual representations. *Proc. ICML*, 2020, pp. 1597–1607.
4. Devlin J., Chang M.-W., Lee K., Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proc. NAACL*, 2018, pp. 4171–4186. doi: 10.18653/V1/N19-1423.
5. Jia C., Yang Y., Xia Y., Chen Y.-T. et al. Scaling up visual and vision-language representation learning with noisy text supervision. *Proc. ICML. PMLR*, 2021, pp. 4904–4916.
6. Zhou K., Yang J., Loy C.C., Liu Z. Learning to prompt for vision-language models. *IJCV*, 2022, vol. 130, no. 9, pp. 2337–2348. doi: 10.1007/s11263-022-01653-1.
7. Zhou K., Yang J., Loy C.C., Liu Z. Conditional prompt learning for vision-language models. *Proc. IEEE Conf. CVPR*, 2022, pp. 16816–16825. doi: 10.1109/CVPR52688.2022.01631.
8. Минченков М.В., Хельвас А.В., Юрин Д.В. Алгоритм автоматической сегментации изображений для систем технического зрения // *Обработка информации и моделирование: сб. тр.* 2002. С. 1–15.
9. Kirillov A., Mintun E., Ravi N., Mao H., Rolland C., Gustafson L. et al. Segment anything. *Proc. IEEE/CVF ICCV*, 2023, pp. 4015–4026. doi: 10.1109/ICCV51070.2023.00371.
10. Madhuanand L., Nex F., Yang M.Y. Self-supervised monocular depth estimation from oblique UAV videos. *ISPRS J.*, 2021, vol. 176, pp. 1–14. doi: 10.1016/j.isprsjprs.2021.03.024.
11. Zhai X., Mustafa B., Kolesnikov A., Beyer L. Sigmoid loss for language image pre-training. *Proc. IEEE/CVF ICCV*, 2023, pp. 11975–11986. doi: 10.1109/ICCV51070.2023.01100.

Automated drone image making using pre-trained artificial intelligence models

Nikita A. Bessarabov ^{1✉}, Aleksey A. Sapozhnikov ¹, Dmitry V. Tatarnikov ¹,
Rushan R. Tyugunov ¹, Aleksandr M. Tsyganov ¹

¹ INPO “KRR “Moy Rayon”, Moscow, 125009, Russian Federation

For citation

Bessarabov, N.A., Sapozhnikov, A.A., Tatarnikov, D.V., Tyugunov, R.R., Tsyganov, A.M. (2025) ‘Automated drone image making using pre-trained artificial intelligence models’, *Software & Systems*, 38(2), pp. 251–260 (in Russ.). doi: 10.15827/0236-235X.150.251-260

Article info

Received: 07.06.2024

After revision: 28.08.2024

Accepted: 09.09.2024

Abstract. The paper discusses an image-mapping method for drone images using pre-trained artificial intelligence models for subsequent training of different class object detectors. The approach consists of three main steps: image segmentation, attention zone extraction, and verifying the segments for the target class using a majority committee of multimodal models. The authors use the Segment Anything Model for object segmentation. Since there can be a significant number of segments, the authors propose an algorithm for building a hierarchical structure over a set of segments. It means that if a segment

is in another segment, then this segment will be nested in the segment containing it. Further processing has the following logic for significant acceleration: if the target class is not found in the parent segment, it will not be found in the nested segments. The authors propose an algorithm for finding attention zones as the most probable areas for finding target class objects in the second stage. For this purpose, they use the pre-trained SigLIP model. All segments from the processed image set are ranked by decreasing probability of finding the target class in them using attention zones. The authors apply the majority committee of three multimodal chatbots (LLaVA, CogVLM, and Mini-Gemini) in the third stage. There is a stopping rule for the chatbots to check, to avoid enumerating all segments and thus speed up the image partitioning process. The authors conducted a computational experiment to demonstrate the proposed approach metrics and speed.

Keywords: dataset, image segmentation, large language models, chatbot, neural networks, generative AI, image mapping

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E. (2017) 'ImageNet classification with deep convolutional neural networks', *Communications of the ACM*, 60(6), pp. 84–90.
2. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R. (2020) 'Momentum contrast for unsupervised visual representation learning', *Proc. IEEE Conf. CVPR*, pp. 9726–9735. doi: 10.1109/CVPR42600.2020.00975.
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G. (2020) 'A simple framework for contrastive learning of visual representations', *Proc. ICML*, pp. 1597–1607.
4. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2018) 'Bert: Pre-training of deep bidirectional transformers for language understanding', *Proc. NAACL*, pp. 4171–4186. doi: 10.18653/V1/N19-1423.
5. Jia, C., Yang, Y., Xia, Y., Chen, Y.-T. et al. (2021) 'Scaling up visual and vision-language representation learning with noisy text supervision', *Proc. ICML. PMLR*, pp. 4904–4916.
6. Zhou, K., Yang, J., Loy, C.C., Liu, Z. (2022) 'Learning to prompt for vision-language models', *IJCV*, 130(9), pp. 2337–2348. doi: 10.1007/s11263-022-01653-1.
7. Zhou, K., Yang, J., Loy, C.C., Liu, Z. (2022) 'Conditional prompt learning for vision-language models', *Proc. IEEE Conf. CVPR*, pp. 16816–16825. doi: 10.1109/CVPR52688.2022.01631.
8. Minchenkov, M.V., Khelvas, A.V., Yurin, D.V. (2002) 'Algorithm of automatic image segmentation for technical vision systems', *Proc. Information Processing and Modeling*, pp. 1–15 (in Russ.).
9. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L. et al. (2023) 'Segment anything', *Proc. IEEE/CVF ICCV*, pp. 4015–4026. doi: 10.1109/ICCV51070.2023.00371.
10. Madhuanand, L., Nex, F., Yang, M.Y. (2021) 'Self-supervised monocular depth estimation from oblique UAV videos', *ISPRS J.*, 176, pp. 1–14. doi: 10.1016/j.isprsjprs.2021.03.024.
11. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L. (2023) 'Sigmoid loss for language image pre-training', *Proc. IEEE/CVF ICCV*, pp. 11975–11986. doi: 10.1109/ICCV51070.2023.01100.

Авторы

Бессарабов Никита Анатольевич¹,

к.т.н., руководитель

проектного офиса,

n.bessarabov@krrmr.ru

Сапожников Алексей Александрович¹,

зам. начальника отдела проектного офиса,

a.sapozhnikov@krrmr.ru

Татарников Дмитрий Владимирович¹,

зам. руководителя проектного офиса,

d.tatarnikov@krrmr.ru

Тюгунов Рушан Рашидович¹,

аналитик данных отдела проектного офиса,

rushan.tyugunov@gmail.com

Цыганов Александр Михайлович¹,

зам. начальника отдела проектного офиса,

a.tsyganov@krrmr.ru

Authors

Nikita A. Bessarabov¹,

Cand. of Sci. (Engineering),

Head of Project Office,

n.bessarabov@krrmr.ru

Aleksey A. Sapozhnikov¹,

Deputy Head of Data Modeling Department,

a.sapozhnikov@krrmr.ru

Dmitry V. Tatarnikov¹,

Deputy Head of Project Office,

d.tatarnikov@krrmr.ru

Rushan R. Tyugunov¹,

Data Analyst of Project Office Department,

rushan.tyugunov@gmail.com

Aleksandr M. Tsyganov¹, Deputy Head

of Project Office Department,

a.tsyganov@krrmr.ru

¹ АНО «КРР «Мой район»,
г. Москва, 125009, Россия

¹ INPO “KRR “Moy Rayon”,
Moscow, 125009, Russian Federation