ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

Отчет о командном программном проекте на тему:

Сервис описания изображений для незрячих людей

(промежуточный, этап 1)

Выполнили студенты:

группы БПМИ213, 3 курса	Рябков Игорь Дмитриевич
группы БПМИ213, 3 курса	Курдун Мария Андреевна
группы БПМИ213, 3 курса	Стамбеков Алмасбек Азатбекович
группы БПМИ211, 3 курса	Коротков Антон Сергеевич

Принял руководитель проекта:

Рогачев Александр Игоревич
Штатный преподаватель
Факультет компьютерных наук НИУ ВШЭ

Москва 2024

Содержание

Аннотация

В данной курсовой работе описывается подготовка моделей для создания сервиса, направленного на ассистирование слабовидящим людям. Разрабатываемый сервис предназначен для описания визуального контента, с которым слабовидящие люди сталкиваются в повседневной жизни, и будет предоставлять текстовое описание для загружаемых изображений. Для решения данной задачи мы используем нейросетевые модели компьютерного зрения (CV) и обработки естественного языка (NLP) . В частности, подготавливаются модели для детектирования и распознавания текста на изображениях, для выявления и классификации элементов интерфейса, определения возраста людей на фотографии и генерации описаний сцен, представленных на изображениях. Результаты этих моделей объединяются в единую текстовую аннотацию, которая будет предназначенную для пользователя. Наш проект способствует продвижению принципов цифровой инклюзивности, делая технологии доступнее для слабовидящих людей.

Ключевые слова

Машинное обучение, нейронные сети, компьютерное зрение, обработка естественного языка, распознавание текста, детекция объектов, описание изображений

1 Введение

На сегодняшний день достаточно существенный процент населения не имеет мозможности потреблять медиа контент, из-за проблем со зрением. Поэтому мы решили помочь таким людям получать информацию и управлять своей жизнью с помощью других органов чувств, в частности, слуха. К нам пришла мысль создать мультимодальный сервис, который бы помогал ориентироваться в Интернет-пространстве и описывал происходящее на изображении.

С целью улучшения практической значимости нашего проекта, мы пригласили слабовидящего эксперта помочь нам. Он будет консультировать нас в течение всего производственного процесса. Это достаточно важно, так как мы не всегда в состоянии оценить, что будет важно нашим потенциальным пользователям. Например, в некоторых случаях форма для них важнее, чем цвет.

Планируется, что сервис будет иметь структуру, показанную на Рисунке 1.1

На вход модели будет поступать информация об интерфейсе. Мы будем использовать модель для разбиения скриншотов на блоки: текст, изображение, кнопки и тд - которые мы будем обрабатывать отдельно:

- Если данный блок изображения соответствует классу: Текст - Он будет передаваться на вход OCR модели, которая будет его распознавать и сохранять в символьном виде
- Если данный блок соответствует классу: Изображение - он передаётся на вход классификатору доменов[Эта модель дополительна, однако в перспективе её точно хотелось бы видеть в нашем сервисе]. Данный блок будет контролировать, какие типы изображения возможно описать, а какие нет, например, мемы - один из таких непосильных доменов.
  - В случае, если изображение возможно описать, оно передаётся на вход Image Captioning модели, которая генерирует необходимое описание
  - Отдельно запускается модель для поиска лиц на изображении, для их дальнейшего анализа:
    * Найденные лица обрабатывает модель для определения возраста и гендера человека, наш эксперт подчеркнул важность точного определения этих фактора, поэтому на данной задаче был поставлен отдельный акцент
  - Сгенерированное описание соединяется с информацией о возрасте и гендере.

Вся информация, сгенерированная на предыдущих этапах, соединяется в единое описание либо с помощью отдельно обученной под эту задачу модели, либо с помощью заготовленных шаблонов, к примеру:

- "В центре экрана изображено [caption + age + gender] с подписью [text]";
- "Снизу находится картинка с мемом, к сожалению, пока его описать не представляется возможным".

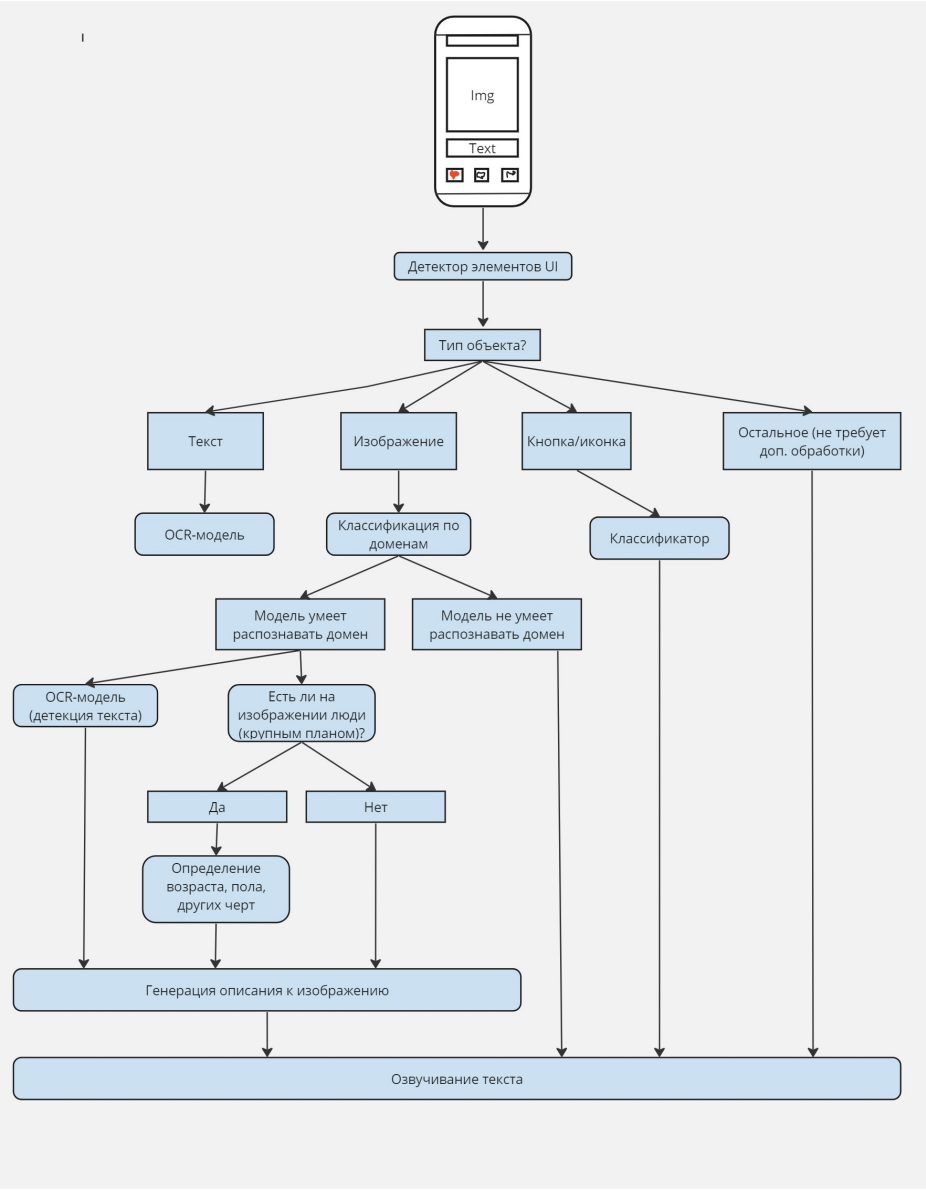

Рис. 1.1: Примерная архитектура сервиса

Сгенерированное описание будет озвучено с помощью TTS (text-to-speech) модели.

1.1 Распределение задач

Модульность данной архитектуры позволила без труда разделить обязанности:

- Антон Коротков занимается первым этапом архитектуры:его задача - найти и дообучить модель разделять интерфейс изображения на блоки, описанные выше
- Мария Курдун занимается подготовкой модели детекции и распознавания текста
- Игорь Рябков на данный момент решает задачу описания изображений
- Алмас Стамбеков изучает модели для определения возраста и гендера по фотографии

В итоге мы хотим разработать сервис, который будет в состоянии считывать информацию с экрана телефона, описывать её и, в конце концов, озвучивать.

2 Анализ существующих решений

Перед началом разработки нашей командой было исследовано, существуют ли похожие решения указанной задачи.

Как оказалось, полноценных онлайн-сервисов для генерации описаний к изображениям с их озвучиванием на данный момент не очень много. Более того, консультантом команды, который сам пользуется аналогичными сервисами, было отмечено, что уже существующие онлайн-сервисы - visionbot[23] и им подобные - в большинстве своем экспериментальные и незаконченные. Существующие онлайн-сервисы:

- пока не способны генерировать оптимальные описания - они получаются слишком многословными и переполненными деталями, а иногда - просто некорректными.
- не предлагают удобный интерфейс для пользования. Например, visionbot требует от пользователя прислать ему либо скриншот для получения описаний, либо ссылку на страницу, которую необходимо описать. Но слабовидящему человеку может быть сложно и неудобно каждый раз копировать изображения или ссылку и передавать эту информацию сервису.

Кроме онлайн-сервисов, полностью направленных на решение рассматриваемой задачи, существуют решения на базе ОС или приложений. В частности, функционал, необходимый для решения описанной задачи, встречается у некоторых виртуальных ассистентов.

Анализ показал, что в различных приложениях подобные инклюзивные функции для слабовидящих людей встречаются достаточно редко. В частности, командой на наличие таких функций были исследованы популярные социальные сети, мессенджеры и браузеры, как приложения, генерирующие наибольший медиатрафик. Нами рассматривались приложения, наибольшим образом использующиеся в RU-сегменте - VK (соцсеть), Telegram (мессенджер), так и приложения, которые пользуются популярностью во всем мире, - WhatsApp (мессенджер), Google Chrome (браузер), Microsoft Edge (браузер). Как оказалось, во всех рассмотренных приложениях, за исключением браузера Microsoft Edge, нет встроенных функций, направленных на решение описанной задачи. В браузере Microsoft Edge представлено определённое решение задачи - функция "Прочитать эту страницу вслух", но оно справляется с задачей лишь частично. В частности, данная функция ищет и проговаривает только текст, игнорируя изображения и виджеты, причём только тот текст, который помещён в соответствующие HTML-блоки.

На базе ОС существуют более эффективные и полные решения. Так, компания Apple реализовала в операционной системе IOS выпускаемых ей смартфонов IPhone и планшетов IPad функцию VoiceOver для помощи слабовидящим людям. VoiceOver не только способен читать вслух текст на web-страницах и в приложениях, но также умеет предварительно генерировать описания к изображениям, кнопкам и иконкам. Эта функция очень хорошо умеет описывает объекты, которые могут появиться на экране мобильного устройства, причём детализацию описаний можно настраивать, что является несомненным преимуществом. Более того, VoiceOver можно очень гибко настроить: можно менять жесты для управления этим виртуальным ассистентом, голоса, которыми произносится текст и т.д. Также VoiceOver может воспринимать информацию на разных языках, а не только на каком-то одном фиксированном.

При всех сильных сторонах VoiceOver, у данной функции есть несколько ограничений:

- можно пользоваться только владельцам техники компании Apple
- ограниченность памяти, выделяемой на ML-модели, использующиеся в VoiceOver
- достаточно низкая скорость обработки изображений (после обработки изображения и выдачи описания к нему VoiceOver показывал большие задержки, вплоть до 30 секунд. Это очень долго)
- закрытая проприетарная платформа, пока не дающая возможность для дальнейшего развития функции

В нашем решении мы планируем реализовать основной функционал VoiceOver, при этом попытаемся убрать указанные выше ограничения. Мы постараемся повысить скорость и качество распознавания и генерации описаний за счёт использования SOTA-моделей. Планируется, что изображения будут описываться более детально, например, вместо того, чтобы сказать, что на картинке просто изображен человек, мы сможем также предоставлять дополнительную информацию про него, такую, как пол или возраст.

3 Обзор литературы

3.1 Описание изображений

После детекции разных блоков интерфейса, необходимо решить задачу описания изображений. Для решения данной задачи, было принято решение взять модели, построенные на архитектуре трансформера, из-за их лучшего качества. Для знакомства с этой архитектурой была изучена статья "Attention is all you need"[21], в которой данная архитектура была представлена в рамках задачи машинного перевода. Главная особенность трансформера: слой Attention, который позволяет моделировать взаимодействие между различными частями входных данных, учитывая их важность. Это делает трансформеры эффективными для решения задач обработки естественного языка, машинного перевода и других, в том числе и для задачи "Изображение"→ "текст".

3.1.1 Модели

Изучая существующие решения, для дальнейшего сравнения между собой были отобраны следуюшие модели (и их версии):

- GIT-Base / GIT-Large[24] Особенность данной модели заключается в её гибкости, способности решать различные задачи image captioning с использованием только одного кодировщика изображений и декодировщика текста. Этот подход отличается от других работ, где применялись более сложные структуры и внешние модули, такие как детекторы или оптическое распознавание символов.

- Vit-GPT2 / Vit-RuGPT Данная модель состоит из двух других: VIT - модель, которая преобразует изображение в последовательностей токенов, которые потом подаются на вход генеративной модели GPT-2. Последняя преобразует последовательность токенов в предложение на Английском языке

- BLIP-Base / BLIP-Large [9] Архитектура BLIP состоит из мультимодальной смеси кодировщика и декодировщика трансформера, который может выполнять три функциональности: обучение унимодального кодировщика для согласования представлений изображений и текста, обучение кодировщика с использованием дополнительных слоев кросс-внимания для моделирования взаимодействий между изображениями и текстом, и обучение декодировщика для генерации подписей к изображениям. Кроме того, модель использует метод Captioning and Filtering для обучения на шумных парах изображение-текст и фильтрации шумных подписей

3.1.2 Набор данных

При выборе набора данных, был поставлен акцент на его объём картинок и на их "разнообразие", а также кол-во примерных описаний. Более релевантным в нашем случае будет набор данных MS-COCO[13] (123,287 изображений, 5 описаний), который содержит большое кол-во картинок разного типа, кроме того, данный набор данных уже успел зарекомендовать себя в задаче Image Captioning, так как его часто используют для обучения и проверки моделей.

Рассматривались и другие наборы данных, например, Flickr8-30K[6] (8000/30000 изображений, 5 описаний) и SBU1M[3] (1000000 изображений, 1 описаний), в которых собраны изрбражения с фотохостинга Flickr. Однако они содержат в себе слишком качественные изображения, с малым кол-вом шума. Наша задача подразумевает описывание изображений в медиа пространстве, где качество всегда ужимается в угоду производительности, поэтому от данных наборов данных пришлось отказаться.

3.1.3 Метрики

Существует достаточно большое количество методов, каким образом можно измерить качество описаний [4]. Самые базовые - это использовать метрики на подобие BLEU, ROUGH-L. BLEU оцениват сгенерированный текст путем сравнения его с эталонными описаниями через совпадение n-грамм, где оценка варьируется от 0 до 1, указывая на точность сопоставления, в то время как ROUGH-L производит сравнения через наиболее длинную общую подпоследовательность, учитывая при этом и полноту, и точность для измерения общего качества. Однако они в основном проверяют пары (предсказания, ответ) на грамматическое сходство, а не семантическое. Поэтому одинаковые по структуре, но разные по смыслу предложения будут рушить подобные метрики.

Есть более интересные эвристики (например, SPICE и CIDEr), которые пытаются учесть важность и сходство некоторых слов для решения вышеупомянутой проблемы: SPICE внутри строит дерево взаимосвязей между словами, CIDEr же опирается на метод TF-IDF, который закладывает понятие важности слова в зависимости от частоты (чем чаще встречается слово, тем менее оно важно)

Однако ничего не сможет обойти обучаемые метрики:

- BERTscore оперирует эмбеддингами. Так получается, что сами по себе эмбеддинги отлично характеризуют то, что они кодируют, поэтому, сравнивая эмбеддинги слов предсказани и ответа, можно добиться достаточно разумной оценки. Легко выявить только один существенный недостаток - необходимость в разметке

- CLIPscore - является фаворитом, он обучен отображать картинки и описание в одно векторное пространство (Чем релевантнее описание, тем ближе его эмбеддинг будет к эмбеддингу картинки). хорошие стороны этой метрики заключаются в том, что она хорошо справляется с разными доменами данных, даже не имея для них разметки

3.2 Определение возраста и пола человека по изображению

Задача данного раздела заключается в построении модели для распознавания возраста и гендера человека и дальнейшем ее интегрировании. Данная модель будет состоять из двух частей: распознавание лица (возможно и частей тела) и непосредственное определение возраста и гендера человека по полученному изображению.

В целях систематического и эффективного подхода к решению данной задачи был разработан план действий:

1 Поиск и анализ наборов данных

   - Исследование доступных наборов данных для задачи распознавания лиц и определения возраста и гендера.
   - Анализ качества и разнообразия данных в выбранных наборов данных.
   - Выбор подходящих наборов данных на основе их релевантности задаче и качества.

2 Изучение литературы и существующих решений

   - Чтение статей и исследований по теме распознавания лиц, определения возраста и гендера.

- Анализ существующих алгоритмов и подходов, их преимуществ и недостатков.
- Изучение архитектур нейронных сетей, успешно применяемых в похожих задачах.

3 Предварительное сравнение базовых моделей

- Выбор базовых моделей на основе обзора литературы.
- Их построение с уже предобученными весами.
- Подсчет качества каждой модели на выбранном наборе данных.

4 Выбор оптимальных моделей

- Выбор итоговых моделей на основе результатов сравнения.
- Анализ ошибок и определение случаев, в которых модели работают недостаточно хорошо.

3.2.1 Наборы данных

Для задачи детекции возраста и гендера на основе изображений лиц широко используются несколько наборов данных, такие как IMDB-WIKI, UTKFace и Adience. Каждый из этих наборов имеет свои особенности, что делает их подходящими для различных исследований и приложений в данном разделе.

IMDB-WIKI[5] является одним из крупнейших доступных наборов данных для анализа лиц, содержащий более 500 тысяч изображений. Эти изображения собраны из интернет-базы данных фильмов (IMDB) и Википедии, включая метаданные с возрастом и гендером, а также положением частей тела. Его преимущество заключается в большом количестве данных и разнообразие лиц, возрастных групп и этнических принадлежностей, что является идеальным вариантом для обучения и тестирования моделей для определения возраста и гендера.

UTKFace[20] содержит более 20 тысяч изображений лиц с аннотациями возраста, гендера и этнической принадлежности. Данный набор включает в себя широкий диапазон возрастов, начиная от детей до пожилых людей, и представителей разных этнических групп, что позволяет решать задачу распознавания демографических характеристик по лицам.

Adience[19] - набор данных фокусируется на определении возраста и гендера в "неконтролируемых"условиях. Содержит тысячи изображений, собранных "в дикой природе без заранее подготовленных поз и освещения. Данные более приближены к реальным условиям использования, что делает его полезным для проверки робастности моделей.

Для обеспечения максимальной эффективности и универсальности модели распознавания лиц и определения гендера, в нашем проекте будет использоваться комбинация вышеупомянутых наборов данных.

3.2.2 Детекция лица

Детекция лица — это процесс определения местоположения лиц на изображении. Это ключевой этап в системах распознавания лиц, который позволяет дальнейшему алгоритму фокусироваться на области изображения, содержащей лицо. Эффективность и точность детекции лиц напрямую влияет на производительность последующего этапа системы. Ниже будут рассматриваться подходы к решению задачи детекции лица.

Haar Cascade[22] основан на признаках Хаара, которые представляют собой специфические структуры яркости на изображении. Алгоритм использует каскадные классификаторы, которые эффективно отсеивают области изображения, не содержащие лиц, на ранних этапах обработки. Данный алгоритм обеспечивает быстроту работы и низкие требования к вычислительным ресурсам.

MTCNN(Multi-task Cascaded Convolutional Networks)[27] - моедль, использующая каскад из трех сверточных нейронных сетей для обнаружения лиц на разных масштабах изображения и одновременного определения их границ и ключевых точек (например, глаз, носа, рта). Его преимущества заключаются в точности детекции лиц и ключевых точек и способности работать с лицами разных размеров и в различных позах.

YOLO v5 (You Only Look Once)[16] — это алгоритм, основанный на сверточных нейронных сетях, который разбивает изображение на сетку и предсказывает ограничивающие рамки и вероятности классов для каждой ячейки сетки одновременно. YOLO обеспечивает быстродействие в реальном времени с высокой точностью, способность обнаруживать объекты различных классов (включая не только лица, но и, например, тело) на одном изображении.

YOLO v8 - алгоритм, представляющий собой последнее поколение в серии You Only Look Once. Ожидается более эффективное выполнение задачи детекции лица с новой версией.

3.2.3 Модели оценки возраста и гендера

Определение возраста и гендера по изображениям лиц является ключевой в данной разделе. Эффективное решение этой задачи требует разработки алгоритмов, способных точ-

но идентифицировать возраст и гендер человека на разнообразных изображениях. Ниже представлены рассматриваемые модели.

DEX (Deep EXpectation)[18]использует глубокую сверточную нейронную сеть, основанную на архитектуре VGG-16, для прогнозирования возраста человека по его лицу на изображении. Основной идеей является использование регрессии и классификации одновременно для прогнозирования возраста, где возрастные группы используются как классы, и в конечном счете вычисляется ожидаемый возраст как взвешенная сумма по всем классам. Это позволяет модели точно и надежно определять возраст, учитывая разнообразие человеческих лиц. К главными преимуществамм данной модели можно отнести высокую точность за счет глубины и количества обучаемых параметров. Данная модель идет предлагается в паре с набором IMDB+WIKI, так как на момент составления она показала лучший на тот момент результат, что делает ее превосходным примером для сравнения с другими моделями.

MiVOLO (Multi-input Transformer for Age and Gender Estimation) [7] - инновационная модель, направленная на одновременное решение задач определения возраста и гендера по изображениям, включая те, которые не содержат лиц. MiVOLO использует два сверточных стебля (conv-stem) для обработки изображений целиком и отдельных фрагментов (например, лица и тела) независимо друг от друга. Это позволяет модели эффективно извлекать и комбинировать признаки из разных областей изображения без потери детализации. Для объединения признаков на раннем этапе обработки был используется модуль, применяющий механизм cross-attention. Этот подход позволяет сначала обогащать признаки вниманием в одну сторону, затем в другую, и в результате сжимать их через многослойную сеть для достижения целевой размерности признаков. Такая структура обеспечивает объединение информации из разных источников и способствует повышению точности предсказаний.

3.3 Детекция и распознавание текста

Задача — построить и интегрировать модель распознавания текста в пайплайн проекта, с фокусом на текст в естественной среде, например, уличные вывески и этикетки.

3.3.1 Подходы

Мы рассматривали две стратегии в построении таких моделей: end-to-end модели, прямо преобразующие изображение в текст, и двухступенчатый подход, где первая модель выделяет текст на картинке ограничивающими рамками (bounding boxes), а вторая — распознает текст в этих рамках. Из-за сложности обучения и настройки end-to-end моделей, а

также для лучшей гибкости,был выбран подход с двумя моделями — для детекции и распознавания текста.

3.3.2 Набор данных

Для решения задачи требуется набор данных картинок с текстами. Так как наша цель — работать с русским и английским языком, то обычные наборы данных вроде ICDAR, Total-Text и CTW не подходят, так как они используют английский (последний — китайский) языки, поэтому выбран менее известный RusTitW [14] , который включает как синтетические изображения, так и реальные. Синтетическая часть содержит около 600 тысяч картинок со случайными словами на случайных фонах. При генерации такого набора данных на картинке выделяются однородные области,текстом описывается форма произвольной фигуры, случайно выбирается шрифт и размер, настраивается перспектива и текст смешивается с исходной картинкой, чтобы получить как можно более натуральное изображение. Однако это создает риск ухудшения качества работы модели в реальных условиях и возможность неточности метрик. Поэтому реальные данные из другой части RusTitW, насчитывающей более 13 тысяч изображений с русским и английским текстом, будут использованы для точных метрик и потенциального дообучения моделей для повышения качества.

3.3.3 Метрики качества детекции

В качестве подходов для оценивания качества моделей детекции мы рассматривали метрику, использованную для оценки качества на соревновании ICDAR2015, где критерием успешности детекции является Intersection over Union (IoU):

$$IoU(box1, box2) = \frac{\text{area of overlap}(box1, box2)}{\text{area of union}(box1, box2)}$$

Также был происследован метод TedEval [8], который основывается на сопоставлении текстовых экземпляров и символьном уровне оценки. Этот подход включает создание пар между размеченными и предсказанными рамками, требуя, чтобы полнота и точность превышали установленные пороги, и исключает многострочные совпадения на основе анализа углов. Далее, расчеты recall и precision ведутся на символьном уровне, для каждого ground truth подсчитывается доля букв которые были покрыты хотя бы одним bounding box, и это значение усредняется по всем ground truth, а также для каждого bounding box подсчитывается

доля букв,которую он покрыл в сопоставляющихся ground truth, и также усредняется:

$$precision = \frac{\sum_{i=1}^{n} precision_i}{n}$$ где n - количество задетектированных рамок

$$recall = \frac{\sum_{j=1}^{m} recall_j}{m}$$ где m - количество ground truth рамок

$$precision_i = \frac{\sum_{j=1}^{m} \sigma_{ji} \cdot \text{word match}_{ji}}{\sum_{j=1}^{m} \sigma_{ji} \cdot len_j}$$

$$recall_j = \frac{\sum_{k=1}^{len_j} I[(\sum_{i=1}^{n} \sigma_{ji} \cdot m_{ji}^k) > 0]}{len_j}$$

$$\sigma_{ji} = I[\text{ j-ый ground truth в паре с i-ым bounding box }]$$

$$\text{word match}_{ji} = \text{количество букв в j-ом ground truth, покрытые i-ым bounding box}$$

$$len_j = \text{количество букв в j-ом ground truth}$$

$$m_{ji}^k = I[\text{ k-ая буква в j-ом ground truth покрыта i-ым bounding box }]$$

3.3.4 Модели детекции

Среди изученных моделей детекции текста выделяются регрессионные подходы, которые напрямую предсказывают ограничивающие рамки текст и подходы, основанные на сегментации, архитектуры придерживающиеся данного подхода, обычно комбинируют предсказания на уровне пикселей и алгоритмы пост-процессинга чтобы получить рамки.

Примером регрессионной модели служит EAST [28], эта модель имеет U-net подобную архитектуру, анализируя карты признаков на разных уровнях для обнаружения мелкого и крупного текста, возвращая уровень уверенности и геометрические координаты текстовых областей, включая углы поворота.Для устранения перекрывающихся прямоугольников применяется NMS алгоритм.

Пример модели, придерживающаяся подхода, основанного на сегментации - PSENet[10]. Эта модель сегментирует текст на уровне пикселей с алгоритмом прогрессивного расширения масштаба, предсказывая маски текста разных масштабов, которые объединяются через BFS-подобный алгоритм. Это помогает различать текстовые экземпляры, расположенные близко друг к другу. Еще один пример модели основанной на сегментации - SAST [25]. Она использует point-to-quad метод для сегментации и внедряет два Context Attention Block, где матрица внимания вычисляется для каждого пикселя только по его столбцу и строке для учета глобального контекста. Далее формируются 4 матрицы - бинарная маска для централь-

ной линии текста (TCL), расстояние от верхних и нижних границ текста (TBO), смещение пикселя от центра текста (TCO) и четырехугольники текста (TVO). Далее происходит кластеризация TCL карты – комбинируются знания из TVO и TCO карт и используется NMS алгоритм.

В случае сложной геометрии текста на картинке, у многих моделей может возникнуть проблемы с его детекцией. Эту проблему решает модель FCENet, которая предлагает новый подход в детекции текста, раскладывая контур текста как функцию в ряд Фурье, если предсказывать какое-то количество коэффициентов ряда для низких частот, то с помощью обратного преобразования можно получить хорошую аппроксимацию контура текста. Помимо этой ветви регрессии параллельно идет ветвь классификации : прогнозируется попиксельная маска текстовых областей, а также маска центра текста (это помогает отфильтровать некачественные предсказания около границ) - для того чтобы получить итоговую карту мы перемножаем эти две матрицы. После используется NMS и обратное преобразование Фурье чтобы получить уже непосредственно границы.

Еще одной сложностью моделей построенных на сегментации, является трудный и долгий процесс постобработки для получения итоговых рамок. Эту проблему решает DBNet, внедряя адаптивную бинаризацию, которая упрощает перевод степени уверенности в бинарные метки. Используя аппроксимированную дифференцированную бинаризацию, модель обучается эффективнее, ускоряется постобработка за счет того что бинарные карты хорошо подобраны, улучшается отделение текстовых экземпляров и обработка текста сложной формы. DBNet++ улучшает DBNet, добавляя модуль Adaptive Scale Fusion для адаптивного объединения карт признаков разных масштабов с помощью внимания, таким образом модель становится более устойчивой к масштабу текста, не теряя при этом в качестве.

4 Полученные результаты

4.1 Детекция элементов интерфейса

4.1.1 Задача

Первым элементом решения задачи является разбиение изображения на блоки интерфейса и классификация блоков по типу элемента интерфейса. Обе задачи - локализация объектов и их классификация - могут решаться одновременно. Такую объединённую задачу называют детекцией изображений.

Для того, чтобы успешно решить задачу детекции элементов интерфейса, необходимо

выбрать оптимальную для детекции модель (или алгоритм) машинного обучения и обучить её. Под оптимальностью имеется в виду не только качество инференса модели с точки зрения метрик, но и быстродействие модели при его генерации, чтобы обеспечить общую высокую скорость ответа сервиса.

Для обучения необходимо найти или сгенерировать данные, представляющие различные скриншоты интерфейсов UI, которые пользователи могут видеть на своих электронных устройствах. Итоговый набор данных должен обладать следующими характеристиками:

- не менее 1000 объектов в наборе
- полнота разметки (отсутствие неразмеченных объектов интерфейса)
- аккуратная локализация объектов интерфейса (так, с одной стороны, границы рамок, в которые помещаются объекты, должны полностью покрывать их, с другой стороны, рамка не должна быть избыточна - она должна покрывать исключительно свой объект)
- оптимальное количество классов (не более 20 - чтобы не было таких ситуаций, что тот или иной объект встречается в наборе данных очень редко, в результате чего модели будет сложно “научиться” определять его)
- никакие два классы не должны пересекаться между собой

Обученная модель должна быть перенесена на сервер и интегрирована в сервис.

4.1.2 Выбор набора данных

В ходе работы над проектом была обнаружена и решена следующая проблема: в рассматриваемом домене компьютерного зрения - элементы интерфейса (далее будем называть их UI-элементами) - очень мало пригодных для решения задачи наборов данных. Среди всех рассмотренных нами наборов больше 50% содержали меньше 1000 объектов, чего крайне мало для дообучения модели. Наборы данных более или менее удовлетворительного размера обладали очень низким качеством разметки ввиду сложности разметки данных для задачи детекции.

В итоге было отобрано 3 следующих набора, более или менее удовлетворяющих перечисленным выше критериям:

- WebUI [26]
- mrtoy/mobile-ui-design[15]

- VINS Dataset[1]

Более детальный дальнейший анализ набора данных **WebUI** (которым пришлось заниматься из-за отсутствия документации к нему) показал, что он не подойдёт под задачу распознавания UI-элементов, так как в нём в качестве классов рассматриваются html-объекты разных web-страниц, которых слишком много, из-за чего по ним будет сложно обучить хорошую модель.

Анализ набора данных **mrtoy/mobile-ui-design** показал, что он также не подойдёт для успешного решения задачи из-за достаточно “общего” выбора классов. В частности, - из-за обилия объектов классов ’group’ ("группа") и ’rectangle’ ("прямоугольник"), которые не очень содержательны (ими описывалось группы из нескольких объектов класса "текст"или "изображение"). Также в этом наборе очень много пустых с точки зрения тех, кто делал разметку к данным, зон, на которых на самом деле есть виджеты, которые также было бы полезно определять.

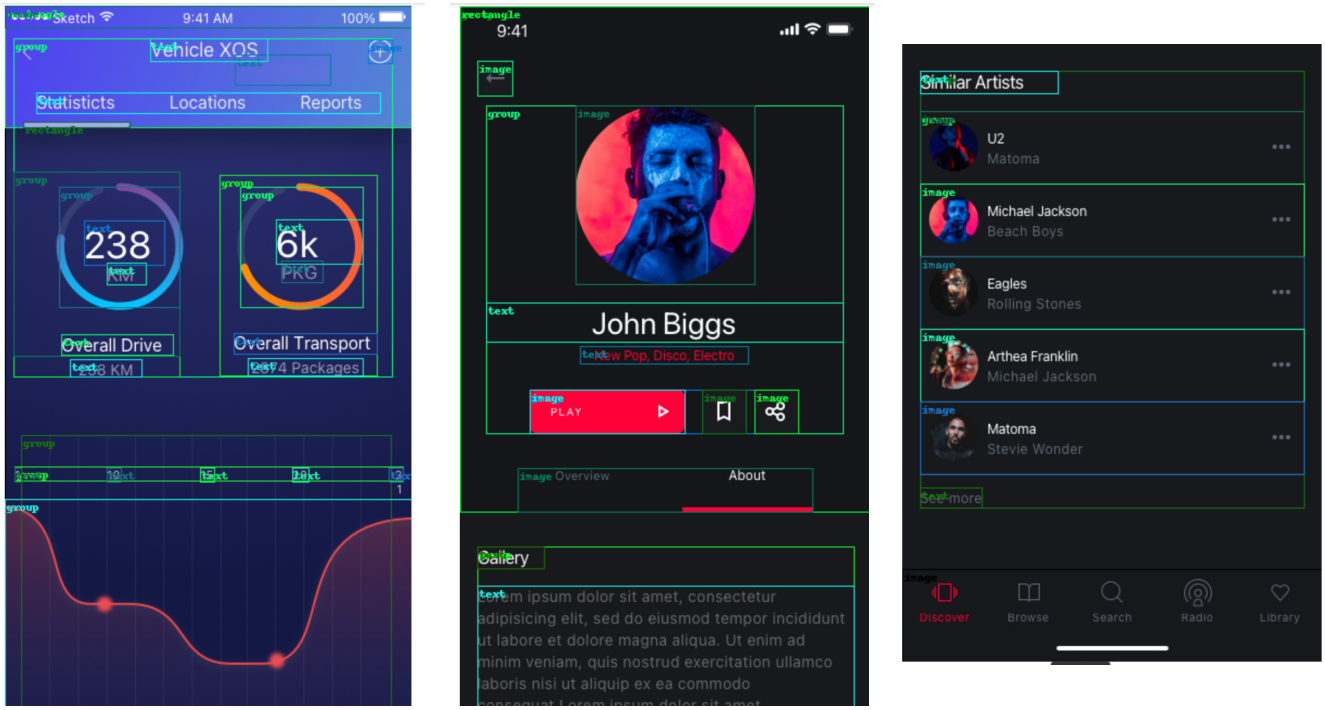

Рис. 4.1: Примеры разметки в наборе данных mrtoy/mobile-ui-design

Набор данных **VINS Dataset** удовлетворяет почти всем указанным выше критериям. В нём достаточно много изображений, разметка выполнена качественно, классы выбраны наиболее оптимально по сравнению со всеми остальными рассматривавшимися наборами. Тем не менее в разметке определённо стоит переработать разделение на классы.

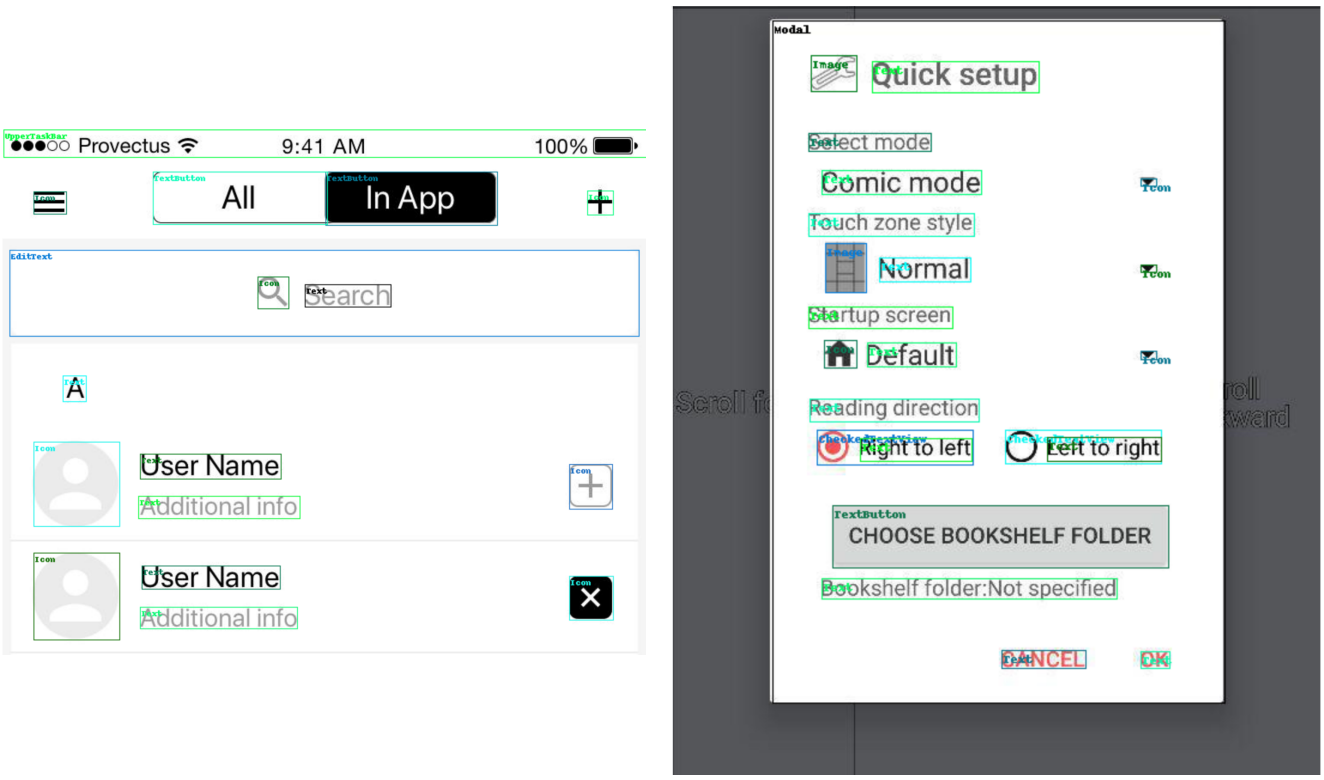

Рис. 4.2: Примеры разметки в наборе данных VINS Dataset

Таким образом, на данный момент найден набор данных, удовлетворяющий всем критериям качества, описанным выше. В связи с этим также было принято дополнительное решение отказаться от генерация дополнительных данных, так как уже существующие объекты достаточно высокого качества и скорее всего синтетические данные в этой задаче лишь сделают распределение объектов более шумным.

Уже проведён первичный анализ выбранного набора данных, удалены поврежденные файлы и описаны конкретные замечания к структуре набора, с учётом которых он будет дорабатываться (в частности - какие именно классы стоит удалить, какие - объединить и т.д.).

4.1.3 Бенчмарк для сравнения моделей

Помимо работы с данными, уже была изучена литература по задаче object detection и на основе изученного материала были приняты решения для создания бенчмарка, приведенные ниже.

Так, выбраны метрики качества модели. Основная используемая метрика в object detection - mAP (mean average precision). Мы будем рассматривать её вариации - mAP50 и mAP@[50-95]: в первом варианте порог равен 0.5, во втором варианте оценка усредняется по всем порогам в диапазоне от 0.5 до 0.95 с шагом 0.05. Также, как было указано выше, крайне

важно обеспечить не только качество детектора, но и его быстродействие, так как детекция - лишь первый этап решения общей задачи. Поэтому кроме mAP будет учитываться inference latency (время на генерацию предсказания для одного объекта).

Также были выбраны следующие модели для экспериментов: Faster RCNN, YOLO v5, YOLO v8, DETR (именно классическая модель, предложенная авторами), CO-DETR.

Faster RCNN[17] - единственная двухэтапная модель детекции в бенчмарке (сначала локализует объекты, затем классифицирует то, что попало в рамки, которыми выделяем объекты). Ожидается, что она покажет достаточно высокое качество, но может оказаться медленнее других моделей, поэтому необходимо провести дополнительные эксперименты.

YOLO v5 - проверенно рабочая и сбалансированная модель. Предполагается, что она покажет один из лучших результатов по балансу между качеством и скоростью. При этом модель обладает обширной документацией, что позволит более тонко её настроить.

YOLO v8 была выбрана как самая новая модель семейства архитектур YOLO. Ожидается, что она покажет очень хорошие результаты, лучше, чем результаты YOLO v5. При этом модель новая, выпущенных по ней статей ещё нет, поэтому требуется дополнительная проверка этой гипотезы.

Также были выбраны модели семейства **DETR**[2] (Detection Transformer), так как модели этого семейства показывают очень качественные результаты. В частности, модель CO-DETR [30] является SOTA-моделью в области детекции на наборе данных COCO. При этом есть риски, что из-за особенностей окружения и отсутствия документации модель может не запуститься. Модель DETR была выбрана для наглядного сравнения с CO-DETR.

4.2 Описание изображений

4.2.1 Задача

После классификации элементов интерфейса на блоки, появляется задача описания тех из них, что представляют из себя фотографии и картинки. Данная проблема не нова, однако в контексте нашей задачи есть несколько существенных нюансов, которые нужно учесть:

- Сервис должен работать в реальном времени, а значит модель должна выдавать ответ за доли секунды для достижения положительного пользовательского опыта
- Описательные модели в том виде, в котором они есть сейчас, не применимы, так как они обучались преимущественно для зрячих пользователей, а значит, иногда их ответ

Model	CLIPscore on COCO	Time (ms)	Parameters cnt
VIT-RuGPT	0.1831	755	1074584064
GIT-Base	0,2715	183	176619066
BLIP-Base	0.2898	230	247414076
VIT-GPT2	0.2917	161	239195904
GIT-Large	0.2723	1513	394196026
GIT-Large-coco	0.3085	1490	394196026
BLIP-Large	0.3054	370	469732924

Таблица 4.1: Сравнение моделей

будет не релевантен (Например, значение цвета объекта часто не так важно, в отличие от его формы)

- Разнообразие доменов изображений, которые необходимо охватить, огромно. Нужно убедиться, что модель хорошо работает с разными типами контента, а также уметь обрабатывать материал, с которым модель справляется плохо.

Процесс изучения был разделён на этапы:

- Изучить домены, которые могут встретиться в медиа
- Подобрать набор данных, который содержит большое разнообразие картинок
- Выбрать метрики для оценки качества моделей
- Изучить какие существуют готовые модели для решения подобной задачи
- Выбрать модель(-и), которая(-ые) хорошо решает(-ют) поставленную задачу
- Найти слабые места модели

4.2.2 Сравнение моделей

В Таблице 4.1 представлены результаты моделей, которые были посчитаны на тестовой выборке набора данных MS COCO, за основную метрику был взят CLIP.

Русскоязычная модель VIT-RuGPT точно не подходит под нашу задачу, она показала плохое значение метрики, а также имеет просто недопустимо большое кол-во параметров

Изначально предполагалось сделать акцент на модели базового типа, так как они не только имеют достаточно хорошее значение метрик, но и более высокую производительность. Однако, исследовав их качество на доменах, которые не встречаются в наборе данных MS-COCO, было выявлено, что они работают на них достаточно нестабильно. Примеры можно увидеть ниже в блоке ”Домены”, где показаны некоторые описания модели BLIP-Base

GIT-Large показала достаточно неплохое качество на тестовой выборке, однако имеет очень высокое время одного предсказания. Учитывая небольшое число параметров, данный факт является парадоксальным. На данный момент не удалось выявить причину этой аномалии

Наиболее подходящий для нас результат показала модель BLIP-Large, было принято решение остановиться на ней (не только из-за значения метрики, но и из-за большей стабильности на различных доменах)

Было бы неправильно не упомянуть более современную версию BLIP: BLIP-2 однако она имеют слишком большое число параметров, даже у самой легковесной её версии их больше 3 млрд, что не допустимо в контексте нашей задачи

4.2.3 Домены

Изображения в медиа можно разбить на следующие домены:

1 Фотографии, пейзажи

2 Мультяшные картинки, аниме

3 Текст

4 Портреты

5 Мемы

6 NSFW (18+ контент)

С первым доменом всё достаточно хорошо, рассмотренные модели показали достаточно приемлемый результат. Стоит только подметить, что более легковесные модели часто упирались в самоповторы, что очень заметно

С мультяшными картинками всё не так однозначно, большинство моделей часто выдавали достаточно странный результат: Все базовые модели работали через раз, Русунок 4.4 яркий тому пример; GIT-Large показал достаточно хорошее качество, однако в его предсказаниях часто содержатся специальные символы, которые неизвестно чем заполнять (заметна тавтология). BLIP-Large напротив показал не только высокое качество описаний, но и лучшую стабильность.

Несмотря на превосходство BLIP-Large, важно отметить ситуации, где ни одна модель не выдала приемлемый результат:

- Недавно придуманные персонажи
- Забытые персонажи

(у модели просто не было примеров в обучающей выборке)

Рис. 4.3

BLIP-Large: a close up of a person holding a gun in a room.
BLIP-Base: a man in a cloak holding a gun
GIT-Large: 0] is a japanese manga and anime character from the anime anime series. he is a japanese

Рис. 4.4

BLIP-Large: venom is a character in the venomverse series.
BLIP-Base: venom venom venom ... (15 раз слово venom)
GIT-Large: venom art print featuring the digital art venom by [unused0]

Рис. 4.5

BLIP-Large: sonic the hedge running in a game.
BLIP-Base: sonic running through the jungle
GIT-Large: sonic the hedgehog game wallpapers

Третий домен покрывать не нужно (задача для OCR модели)

С четвёртым пунктом всё не так однозначно, смотря на портрет, люди в первую очередь хотят видеть эмоции человека, вырожение его лица, детали. Модели описывают их в слишком общем ключе, хотелось бы больше конкретики с фокусом на эмоции, на Рисунках 4.6-4.8 предоставлены некоторые примеры. В данном домене BLIP-Large снова показал себя лучше остальных моделей: У легковесных моделей есть самоповторы; У GIT-Large неудобные специальные символы (и достаточно скучные описания)

Рис. 4.6

BLIP-Large: there is a man in a black coat standing on a city street.

BLIP-Base: a man with a black coat and a black coat

GIT-Large:0]is a spanish model who is best known for his role as [unused0] in

Рис. 4.7

BLIP-Large: there is a woman with a white dress posing in a field.

BLIP-Base: a woman with long hair standing in a field

GIT-Large: portrait of a woman in a field

Рис. 4.8

BLIP-Large: a painting of a woman with long hair and a white top.

BLIP-Base: a painting of a woman with long hair

GIT-Large: portrait of a young woman

Основная сложность заключается в последних двух классах: для описания мемов необходимо понимание большого кол-ва контекста, которого у модели нет; примеров NSFW контента в обучающей выборке мало, поэтому результат, выдаваемый моделью, неприемлемый для нашей задачи.

Для борьбы с проблемой, было решено просто уведомлять пользователя о том, что это домен "мем"без попыток его описания.

4.3 Определение возраста и пола человека по изображению

4.3.1 Сравнения моделей

В рамках исследования ключевым этапом является анализ и сравнение различных моделей. Выбор наиболее подходящей модели представляет собой важную задачу, от решения которой зависит не только точность и надежность предсказаний, но и эффективность внедрения решения в реальные приложения и системы. В данном разделе представлен первоначальный обзор выбранных моделей, оценка их производительности и анализ ключевых

метрик.

Таблица 4.2: Сопоставление предобученных моделей детекции лица на тестовой подвыборке IMDB+WIKI (56087 изображений).

Model	Faces detected	Labels missed	Average time (ms)
Haar Cascade	179664	2992	38.2
MTCNN	128431	105	155.6
YOLO v5	184318	0	40.94
YOLO v8	170236	67	13.52

По предварительным результатам из Таблицы 4.2 выясняется, что YOLO v8 выделяется как самая быстрая модель, в то время как YOLO v5 показывает наилучший баланс между скоростью и точностью обнаружения. MTCNN требует большего времени обработки, плюс к этому имеет пропуски в обнаружении помеченных лиц. Haar Cascade же показывает плохое качество обнаружения. В дальнейшем планируется использовать YOLO v8 в качестве модели обнаружения лица (и при необзодимости других частей тела). Далее рассмотрим модели оценки возраста и пола.

Таблица 4.3: Сопоставление предобученных моделей на тестовой подвыборке IMDB+WIKI (56087 изображений).

Model	Age (MAE)	Gender (Acc)	#params (M)	Average time (ms)
DEX	9.18	85.2	134.7 + 134.2	10 + 9.5
MiVOLO	8.39	99.3	25.86	5.63

Учитывая результаты из Таблицы 4.3, MiVOLO выделяется как более продвинутая модель для задач определения возраста и гендера, предлагая лучшее сочетание точности и скорости обработки по сравнению с DEX. Пример выполнения оценки возраста по изображению с использованием MiVOLO и YOLOv8 изображен на Риунке 4.9. Так, можно наглядно убедиться в качестве выбранных моделей.

4.4 Детекция и распознавание текста

Данная задача заключается в том чтобы построить модель детекции и распознавания текста на картинках. К текущему моменту сделаны следующие части работы:

1 Выбор доменов картинок и поддерживаемых языков для распознавания

2 Выбор набора данных, изучение его содержимого

3 Изучение архитектур различных моделей детекции текста

4 Ознакомление с фреймворками для работы с моделями детекции и моделями распознавания текста

Рис. 4.9: Пример работы MiVOLO. В данном примере детекция лица и тела происходит с помощью YOLO v8.

5 Выбор и реализация метрик для оценки качества моделей детекции, уточнение набора доменов для визуальной оценки

6 Анализ работы моделей детекции с учетом времени инференса, выделение проблемных доменов

7 Выбор метрик для оценки качества моделей распознавания

4.4.1 Метрики детекции

Вначале мы немного модифицировали метрику, использованную для оценки качества на соревновании ICDAR2015, основанной на метрике Intersection over Union (IoU).Считается, что предсказанный bounding box соответствует ground truth box, если их коэффициент IoU больше определенного порога (в нашем случае 0.5). Однако из-за того что набор с реальными данными имеет очень завышенные рамки, которые могут охватывать по несколько строк текста и имеют большие зазоры между границами и самим текстом, часть IoU, даже при хороших предсказаниях, не превышает этот порог, поэтому было добавлено еще одно условие для покрытия как можно большего числа положительных случаев. Предсказанный bounding

box теперь считается соответствующим ground truth, если происходят следующие условия:

$$\begin{cases} IoU(\text{bounding box, ground truth}) > 0.5 \\ \begin{cases} IoU(\text{bounding box, ground truth}) > 0.2 \\ \frac{\text{area of overlap (bounding box, ground truth)}}{\text{area (bounding box)}} > 0.9 \end{cases} \end{cases}$$

Такой подход (далее буду обозначать как asis) позволяет учитывать больше правильных совпадений. Исходя из этого, рассчитываются точность (precision), полнота (recall) и f1 для оценки производительности модели:

$$\text{Precision} = \frac{\text{количество детекций, имеющих соответствие с ground truth}}{\text{количество всех детекций}}$$

$$\text{Recall} = \frac{\text{количество ground truth, имеющих соответствие с предсказанными рамками}}{\text{количество ground truth рамок}}$$

$$F1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Также был происследован метод TedEval [8] для оценки качества детекции текста. В результате анализа выяснилось, что TedEval неэффективен для наших данных с многострочными и завышенными ground truth рамками и эта метрика сильно занижалась из-за преждевременного исключения хороших детекций. (Рисунок 4.10) В ответ на это была разработана дополнительная метрика (далее буду обозначать как join), объединяющая пересекающиеся bounding boxes, чтобы лучше адаптироваться к неточностям разметки и предоставить более реалистичную оценку.

4.4.2 Фреймворки

Были изучены фреймворки MMOCR и PaddleOCR, которые содержат инструменты для обучения моделей детекции и распознавания текста, а также содержат уже обученные модели (в основном на популярных англоязычных наборах данных и на синтетически сгенерированных наборах данных). Были рассмотрены разные модели детекции для определения их эффективности на реальных данных через описанные метрики, визуальное качество по доменам и время инференса. Домены включали декоративный шрифт, дорожные знаки, светящиеся вывески, простые и размытые тексты, этикетки, обложки книг, инструкции, текст на всю картинку, изогнутый и вертикальный текст.

Рассмотренные модели детекции : EAST, SAST, PSENet, FCENet, DBNet, DBNet++

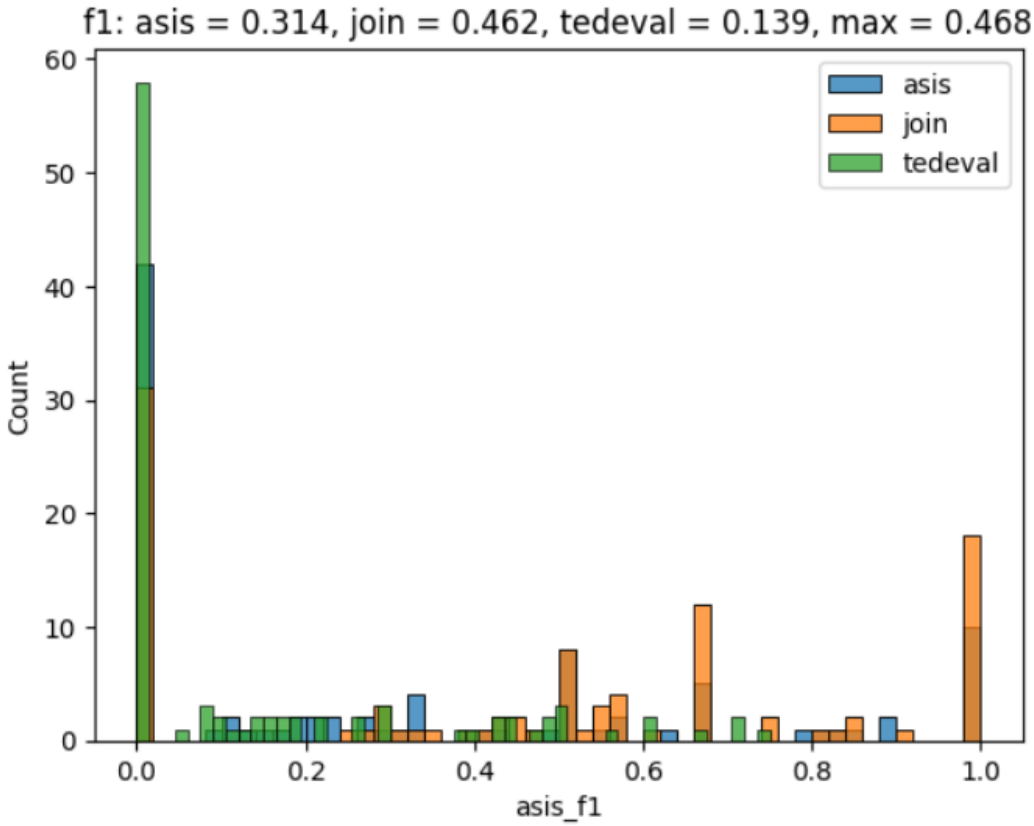

Рис. 4.10: Распределение оценок качества детекции модели DBNet++ на части картинок из набора с реальными данными. На графике представлена метрика f1 score, посчитанная с помощью 3 подходов к оценке качества детекции : asis, join, tedeval. Max соответствует максимальной метрике среди этих 3 подходов для каждой картинки. В заголовке графика отмечено среднее значение f1 score для каждого из подходов

4.4.3 Модели детекции

В процессе работы были происследованы различные архитектуры и backbones, такие как ResNet50, ResNet18, MobileNet, ResNet50 DCN, и ResNet50 OCLIP. Для каждой модели были выделены проблемные домены, посчитаны и проанализированы метрики качества и время инференса.

EAST [28]: несмотря на быстрое время инференса, генерирует много лишних прямоугольников и показывает слабые результаты на сложных текстах, включая декоративный шрифт, размытый, кривой текст и обложки книг.

PSENet [10]: сталкивается с трудностями при обработке многих доменов и при этом имеет инференс в 15 раз дольше, чем EAST.

SAST [25]: модель эффективна в детекции, но имеет большое время инференса, причем испытывает небольшие сложности с изогнутым и крупным текстом.

DBNet [11]: модель показывает быстрый инференс, хотя результаты ухудшаются с тяжелыми backbone, например, ResNet50.В целом общие проблемные домены это кривой,

большой и крупный тексты.

FCENet [29]: несмотря на уникальность идеи, модель показала низкую эффективность по всем учтенным доменам.

DBNet++ [12] : стала лучшей моделью по балансу качества и времени инференса, при resnet50 dcn в качестве backbone работает неплохо во всех доменах, имея лишь небольшие неточности при кривом тексте. Примеры детекции этой модели представлены на Рисунке 4.11

В Таблице 4.4 отражены метрики качества и время инференса для каждой из протестированных моделей

Model	asis_precision	asis_recall	asis_F1	join_precision	join_recall	join_F1	tedeval_precision	tedeval_recall	tedeval_F1	time
dbnet_resnet18_icdar	0.242132	0.426680	0.286353	0.269890	0.429117	0.309606	0.112681	0.202615	0.134646	76.723676
dbnet_resnet50_dcn_icdar	0.227384	0.396745	0.266969	0.263745	0.410584	0.297068	0.108481	0.190677	0.128901	193.85942
dbnet_resnet50_oclip_icdar	0.233091	0.437170	0.283517	0.305442	0.485434	0.347689	0.102856	0.200294	0.128329	182.492468
dbnet++_resnet50_dcn_icdar	0.234399	0.427504	0.279052	0.280181	0.453935	0.320557	0.118738	0.211138	0.141936	213.233288
dbnet++_resnet50_oclip_icdar	0.223679	0.439496	0.275083	0.277439	0.482130	0.330208	0.096932	0.196260	0.122024	280.574773
psenet_resnet50_icdar	0.083661	0.252509	0.112453	0.174554	0.290955	0.200461	0.034783	0.106134	0.048004	625.222996
psenet_resnet50_oclip_icdar	0.155343	0.362878	0.194330	0.229235	0.413594	0.265822	0.070807	0.155209	0.087955	689.422507
fcenet_resnet50_icdar	0.185233	0.275053	0.199413	0.260991	0.344200	0.274986	0.059754	0.113410	0.069812	883.128599
fcenet_resnet50_oclip_icdar	0.177289	0.319464	0.207192	0.279048	0.409428	0.308898	0.073164	0.131814	0.085464	913.964442
sast_resnet50_icdar	0.234840	0.461668	0.288516	0.318749	0.519200	0.368256	0.110660	0.216503	0.136775	1538
east_resnet50_icdar	0.223178	0.365777	0.253172	0.298379	0.429480	0.326257	0.102065	0.158780	0.114280	41
east_mobilenet_icdar	0.245114	0.370261	0.261164	0.332197	0.422688	0.341748	0.111455	0.165345	0.121754	27

Таблица 4.4: Качество и время инференса для каждой из протестированных моделей. Качество оценивается с помощью 3 метрик: precision, recall и f1 score - по каждому из 3 подходов: asis, join,tedeval.

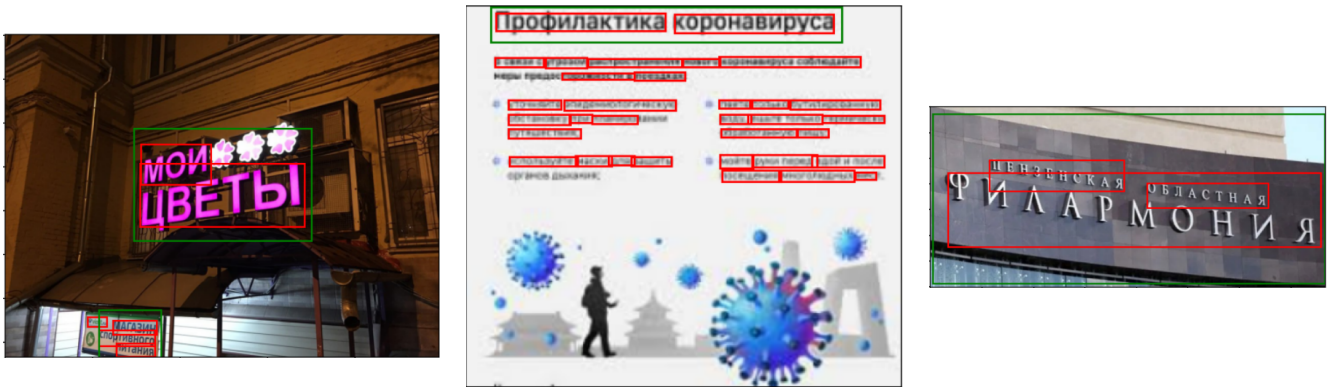

Рис. 4.11: Примеры детекции DBNet++ с backbone resnet50 dcn. Зеленые рамки - ground truth, красные рамки - детекция

4.4.4 Метрики для распознавания

Для оценки модели распознавания текста используем точность по словам, учитывая различные вариации учета регистра, наличия цифр и знаков. В качестве второй метрики применяем NED, отражающую качество с учетом сложности восстановления реального тек-

ста по выходу модели:

$$\text{NED(predicted text, real text)} = \frac{\text{levenshtein distance(predicted text,real text)}}{\text{max(len(predicted text),len(real text))}}.$$

5 Дальнейшая работа

5.1 Детекция элементов интерфейса

В качестве следующих шагов будет закончена предобработка набора данных, будет проведено сравнение моделей с выбором оптимального решения. В дальнейшем будет осуществлено дообучение модели и её интеграция в сервис.

Ниже представлен более детальный план дальнейших действий:

1 **Выбор оптимальной модели**

1.1 Fine-Tuning и Linear Probing моделей и методов из бенчмарка

1.2 Настройка гиперпараметров моделей, проведение дополнительных экспериментов

1.3 Сравнение результатов и выбор оптимального с точки зрения метрик качества метода

2 **Подготовка модели к промышленному использованию**

2.1 Дообучение модели

2.2 Дополнительные эксперименты для повышения качества работы модели

2.3 Загрузка модели на производственный сервер

В ходе дальнейшей работы также следует учитывать следующие потенциальные проблемы:

- Данных в VINS Dataset может оказаться недостаточно для точного определения объектов каждого из классов элементов UI. Особенно это может быть критично для Fine-Tuning'а трансформеров (они требуют очень большого количества объектов в ходе обучения). В случае, если данная проблема окажется критичной, решение будет вырабатываться в момент его реализации. Возможно придётся изменить бенчмарк.
- Из-за отсутствия достаточной документации и опыта работы с моделью CO-DETR, возможно, её не удастся использовать. В случае реализации данной проблемы все модели семейства DETR будут исключены из рассмотрения.

5.2 Описание изображений

В ближайшее время необходимо будет:

- Проанализировать на каких примерах BLIP-Large даёт плохой CLIPscore
- Найти наборы данных для доменов, с которыми модель плохо справляется
- Дообучить модель справляться с трудными типами изображений
- Обучить вспомогательную модель для поиска сложных доменов, которые описать не удастся
- В случае необходимости, найти оптимальную длину одного описания, изучить модель FuseCap (Это BLIP-Large, дообученный выдавать более длинные предсказания)

5.3 Определение возраста и пола человека по изображению

В рамках дальнейшей работы над данной задачей предусмотрено несколько направлений для повышения эффективности и точности моделей определения возраста и гендера. В первую очередь, планируется провести тщательную работу с наборами данных, включая аугментацию и очистку данных, а также интеграцию дополнительных наборов данных для обеспечения большей разнообразности обучающих примеров. Особое внимание будет уделено "хвостам"распределения возраста, где текущие модели показывают наименьшую точность, что потребует дообучения существующих моделей на расширенном и уточненном наборе данных. В дополнение к работе с уже известными архитектурами, запланирована разработка собственной модели на базе MobileNet, специально адаптированной под задачи классификации по возрасту и гендеру. После проведения обучения и тестирования новой модели, будет проведено сравнение ее производительности с результатами предыдущих моделей для выбора наиболее оптимальной архитектуры. Завершающим этапом станет интеграция выбранной модели в общую систему.

5.4 Детекция и распознавание текста

1 Изучение архитектур различных моделей распознавания текста

2 Практическое освоение инструментов для разметки данных

3 Разметка части набора с реальными данными

4 Дообучение нескольких моделей, сравнительный анализ качества и времени инференса

5 Эксперименты с разными комбинациями наборов данных (синтетические и реальные) и параметрами дообучения для оптимизации результатов

6 Интеграция и тестирование модели коррекции ориентации текста в общем пайплайне для улучшения результатов распознавания

7 Корректировка модели путем дообучения ее на трудных доменах

В отличие от моделей детекции, обученные на другом языке модели распознавания не получится использовать, просто потому что у модели не будет понятия о русских буквах. Моделей, обученных на русском языке, практически нет, была найдена только одна модель из PaddleOCR, обученная на синтетическом наборе данных. Такая модель потенциально достаточно ограничена в своих возможностях на домене из реальных данных, поэтому в дальнейших планах изучить архитектуры для распознавания текста, попробовать различные методы дообучения этих архитектур (предобученных на английском языке) с использованием комбинаций синтетических и реальных данных, экспериментировать с параметрами дообучения, проанализировать качество полученных моделей, понять, на каких доменах они ошибаются, выбрать лучшую и подкорректировать ее ошибки, дообучив на самых трудных доменах. Также планируется рассмотреть интеграцию модели коррекции ориентации текста, которые представлены в PaddleOCR, в общий пайплайн для улучшения качества распознавания. Однако для того чтобы дообучать или даже просто тестировать модель распознавания на реальных данных нужна хорошая разметка изначальной картинки, так как перед подачей в модель распознавания из картинки вырезаются задетектированные области с текстом. Так как в наборе с реальными данными рамки очень завышены и часто захватывают большие куски текста – что абсолютно расходится с тем, как работают модели детекции и на чем предобучалась модель распознавания, поэтому можно либо брать выходы модели детекции, на которых уверенность значительна (то есть они потенциально качественные), вырезать и размечать только изображенный текст, или же полностью корректировать изначальную разметку набора данных, можно комбинировать оба этих варианта. В любом случае для этого предстоит ознакомиться с удобными инструментами для разметки данных.

Список литературы

[1] Sara Bunian, Kai Li, Chaima Jemmali, Casper Harteveld, Yun Fu и Magy Seif El-Nasr. «VINS: Visual Search for Mobile User Interface Design». В: *arXiv* (2021). arXiv: arXiv:2102.05216v1 [cs.HC].

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov и Sergey Zagoruyko. «End-to-End Object Detection with Transformers». В: *arXiv* (2020). arXiv: arXiv:2005.12872v3 [cs.CV].

[3] Hugging Face Datasets. *SBU Captions Dataset*. https://huggingface.co/datasets/sbu_captions. (дата обр. 13.02.2024).

[4] Othón González-Chávez, Guillermo Ruiz, Daniela Moctezuma и Tania A. Ramirez-delReal. «Are metrics measuring what they should? An evaluation of image captioning task metrics». В: *arXiv* (2022). arXiv: arXiv:2207.01733 [cs.CV].

[5] *IMDB-WIKI: a dataset for age and gender estimation on movie stars*. https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/. (дата обр. 12.02.2024).

[6] Aditya Jain. *Flickr30k*. https://www.kaggle.com/datasets/adityajn105/flickr30k/data. (дата обр. 13.02.2024).

[7] Maksim Kuprashevich и Irina Tolstykh. «MiVOLO: Multi-input Transformer for Age and Gender Estimation». В: *arXiv* (2023). arXiv: arXiv:2307.04616v2 [cs.CV].

[8] Chae Young Lee, Youngmin Baek и Hwalsuk Lee. «TedEval: A Fair Evaluation Metric for Scene Text Detectors». В: *arXiv* (2019). arXiv: arXiv:1907.01227v1 [cs.CV].

[9] Junnan Li, Dongxu Li, Caiming Xiong и Steven Hoi. «BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation». В: *arXiv* (2022). arXiv: arXiv:2201.12086v2 [cs.CV].

[10] Xiang Li, Wenhai Wang, Wenbo Hou, Ruo-Ze Liu, Tong Lu и Jian Yang. «Shape Robust Text Detection with Progressive Scale Expansion Network». В: *arXiv* (2018). arXiv: arXiv:1806.02559v1 [cs.CV].

[11] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen и Xiang Bai. «Real-time Scene Text Detection with Differentiable Binarization». В: *arXiv* (2019). arXiv: arXiv:1911.08947v2 [cs.CV].

[12] Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao и Xiang Bai. «Real-Time Scene Text Detection with Differentiable Binarization and Adaptive Scale Fusion». В: *arXiv* (2022). arXiv: arXiv:2202.10304v1 [cs.CV].

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick и Piotr Dollár. «Microsoft COCO: Common Objects in Context». В: *arXiv* (2015). arXiv: arXiv:1405.0312v3 [cs.CV].

[14] Igor Markov, Sergey Nesteruk, Andrey Kuznetsov и Denis Dimitrov. «RusTitW: Russian Language Text Dataset for Visual Text in-the-Wild Recognition». В: *arXiv* (2023). arXiv: arXiv:2303.16531v1 [cs.CV].

[15] *mobile-ui-design: a dataset for object detection tasks with a focus on detecting elements in mobile UI designs*. https://huggingface.co/datasets/mrtoy/mobile-ui-design/. (дата обр. 01.02.2024).

[16] Joseph Redmon, Santosh Divvala, Ross Girshick и Ali Farhadi. «You Only Look Once: Unified, Real-Time Object Detection». В: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, с. 779—788.

[17] Shaoqing Ren, Kaiming He, Ross Girshick и Jian Sun. «Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks». В: *arXiv* (2016). arXiv: arXiv:1506.01497v3 [cs.CV].

[18] Rasmus Rothe, Radu Timofte и Luc Van Gool. «Deep expectation of real and apparent age from a single image without facial landmarks». В: *International Journal of Computer Vision* 126.2-4 (2018), с. 144—157. DOI: 10.1007/s11263-017-1059-5.

[19] Tal Hassner. *The Adience Benchmark of Unfiltered Faces for Age, Gender and Subject Classification*. https://talhassner.github.io/home/projects/Adience/Adience-data.html. (дата обр. 12.02.2024).

[20] *UTKFace: Large Scale Face Dataset*. https://susanqq.github.io/UTKFace/. (дата обр. 11.02.2024).

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser и Illia Polosukhin. «Attention Is All You Need». В: *arXiv* (2023). arXiv: arXiv:1706.03762v7 [cs.CL].

[22] Paul Viola и Michael Jones. «Rapid Object Detection using a Boosted Cascade of Simple Features». В: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*. Т. 1. IEEE. 2001, с. I—I.

[23] *VisionBot: Сервис распознавания картинок и текста на них*. https://visionbot.ru/. (дата обр. 14.02.2024).

[24] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu и Lijuan Wang. «GIT: A Generative Image-to-text Transformer for Vision and Language». В: *arXiv* (2022). arXiv: arXiv:2205.14100v5 [cs.CV].

[25] Pengfei Wang, Chengquan Zhang, Fei Qi, Zuming Huang, Mengyi En, Junyu Han, Jingtuo Liu, Errui Ding и Guangming Shi. «A Single-Shot Arbitrarily-Shaped Text Detector based on Context Attended Multi-Task Learning». В: *arXiv* (2019). arXiv: arXiv:1908.05498v1 [cs.CV].

[26] *WebUI: A Dataset for Enhancing Visual UI Understanding with Web Semantics*. https://github.com/js0nwu/webui/tree/main?tab=readme-ov-file/. (дата обр. 12.02.2024).

[27] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li и Yu Qiao. «Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks». В: *IEEE Signal Processing Letters* 23.10 (2016), с. 1499—1503.

[28] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He и Jiajun Liang. «EAST: An Efficient and Accurate Scene Text Detector». В: *arXiv* (2017). arXiv: arXiv:1704.03155v2 [cs.CV].

[29] Yiqin Zhu, Jianyong Chen, Lingyu Liang, Zhanghui Kuang, Lianwen Jin и Wayne Zhang. «Fourier Contour Embedding for Arbitrary-Shaped Text Detection». В: *arXiv* (2021). arXiv: arXiv:2104.10442v2 [cs.CV].

[30] Zhuofan Zong, Guanglu Song и Yu Liu. «DETRs with Collaborative Hybrid Assignments Training». В: *arXiv* (2023). arXiv: arXiv:2211.12860v6 [cs.CV].