Сервис описания изображений для незрячих людей

Научный руководитель: Рогачев А.И.

Работу выполнили:

Рябков Игорь 212

Курдун Мария 213

Коротков Антон 211

Стамбеков Алмас 213

Актуальность задачи

- Визуальная информация имеет определяющее значение во всех жизненных сферах.
- Существенная часть населения сталкивается со сложностями при потреблении визуального контента из-за проблем со зрением.
- В связи с этим мы разработали сервис, способный описывать и озвучивать медиа контент

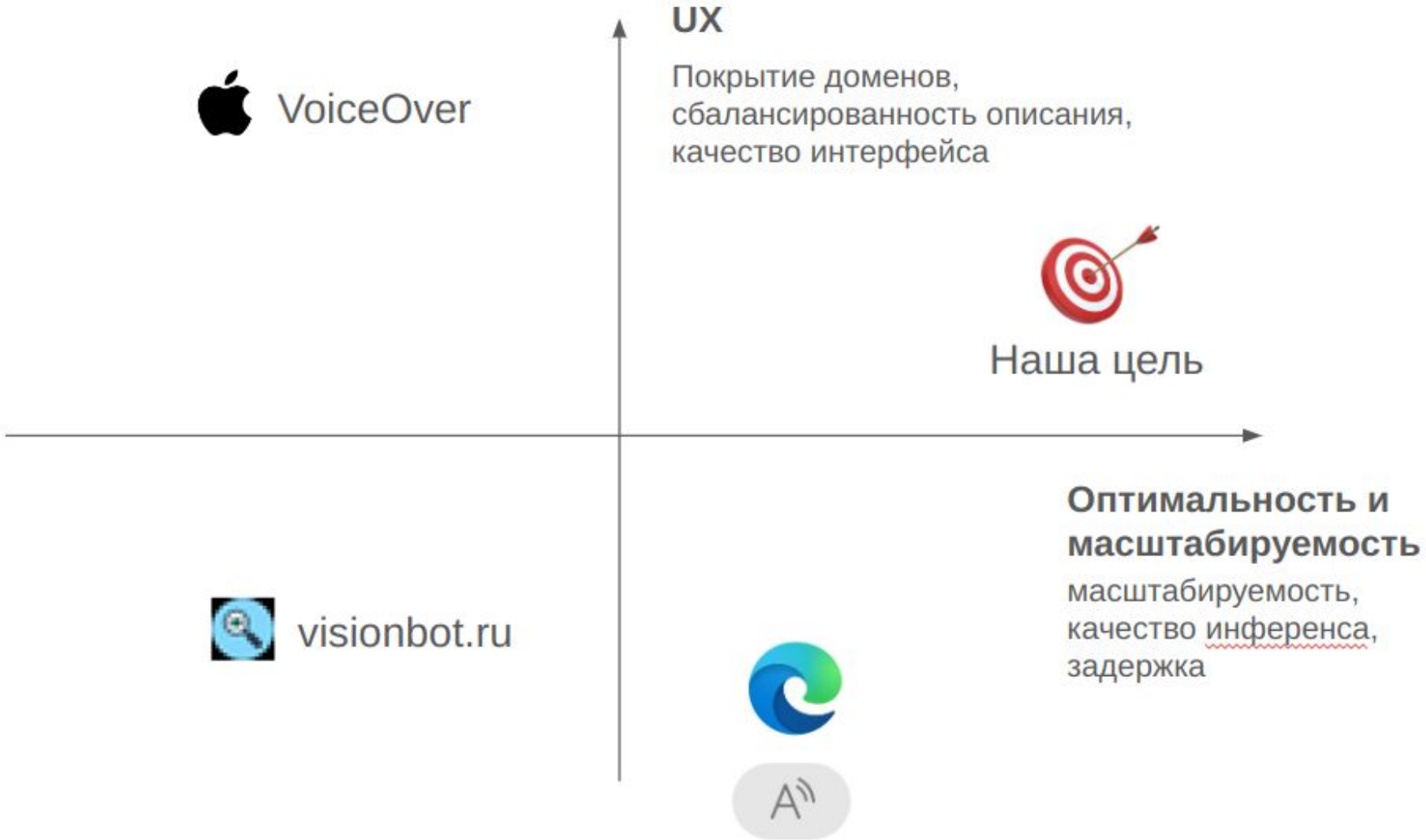
Цель: Описания изображений в медиа пространстве

Задачи каждого из участников:

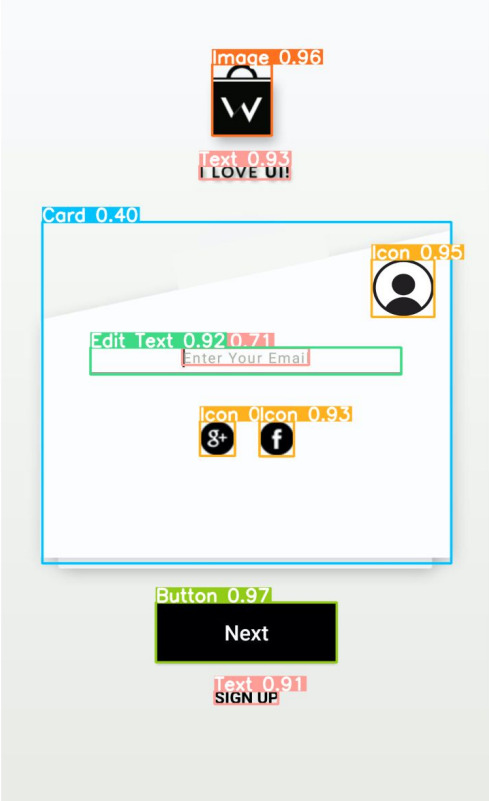
- Антон Коротков - локализация и классификация элементов интерфейса на изображении.
- Мария Курдун - детекция и распознавание текста.
- Алмасбек Стамбеков - определение возраста и пола людей по изображению.
- Игорь Рябков - описание изображений, классификация анимационных и реалистичных картинок.

Общая задача: Разработка мастер-сервиса, который интегрирует работу всех моделей, а также создание обертки для взаимодействия с телеграм-ботом.

Обзор аналогов



Детекция элементов интерфейса: постановка задачи



Для полученного изображения хотим:

- Локализовать на нём все объекты из определённого домена
  - В данном модуле - элементы пользовательских интерфейсов
  - Возможные другие домены: люди, домашние животные и т.п.
- Для каждого локализованного объекта выдать его класс/тип
  - В данном модуле - изображения, кнопки, тексты и т.п.
- Уметь решать эти задачи одновременно

Метрики качества детекции

В качестве метрик качества используем вариации mAP (Mean Average Precision):

- mAP50 (порог IoU = 0.5)
- mAP@[50-95] (усреднение по всем порогам IoU в диапазоне от 0.5 до 0.95 с шагом 0.05)

Также учитываем среднюю скорость инференса модели

$$\mathbf{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

$$\mathrm{mAP} = \frac{1}{C}\sum_{i=1}^{C} \mathrm{AP}_i \quad (C - \text{число классов}, \mathrm{AP}_i - \text{average precision для класса i})$$

Сравнение моделей

Для бенчмарка были выбраны модели семейства YOLO (CNNs):

- YOLOv5 - версия архитектуры, основанная на YOLOv4, с anchor boxes
- YOLOv8 - доработанная YOLOv5, без anchor box’ов и с несколькими heads
- YOLOv9 - модель, основанная на YOLOv7, большой фокус - на уменьшении потерь при передаче информации между слоями

Проблема: модели семейства YOLO - не zero-shot. Zero-shot детекторы показывают более низкое качество по сравнению с обученными под домен моделями.

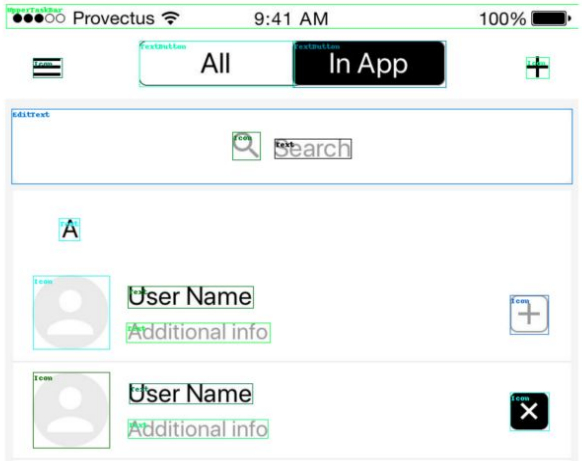
Решения:

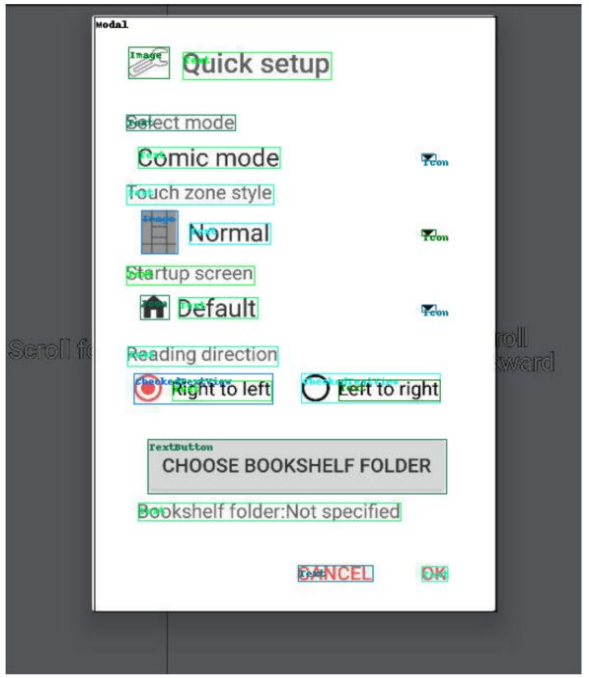
- Полное обучение модели под нужный домен ❌
  - долго
  - ожидаемые результаты хуже, чем при дообучении
- Дообучение из хорошего pre-train’а ✅

Выбор датасета

В ходе работы рассматривались следующие датасеты:

- **WebUI** ❌ - слишком много классов
- **mrtoy/mobile-ui-design** ❌ **-** очень общие классы, много пустых, неразмеченных мест
- **VINS Dataset** ✅ - разметка выполнена качественно, классов - около 20





Доработка датасета

Улучшения:

- Редкие классы были либо удалены, либо объединены с более “общими”
- Были удалены нерелевантные классы, не несущие ценной информации для пользователя
- Были удалены изображения-дубликаты и “битые” файлы

Итог: получили датасет из ~5000 изображений, 14 классов, который готов для дообучения моделей

Как правильно делить выборку?

Проблема: делим выборку из изображений, но работаем с объектами => возможен дисбаланс классов после разбиения между трейном и валидацией

Готовое решение не было найдено

Решение: Разработали собственный алгоритм для стратифицированного разделения выборки на train и val в случае задачи детекции.

- Алгоритм работает за O(len(val) * len(train))
- Использует доработанный функционал разбиения множества в вершине решающего дерева на подмножества в детях

Как правильно делить выборку?

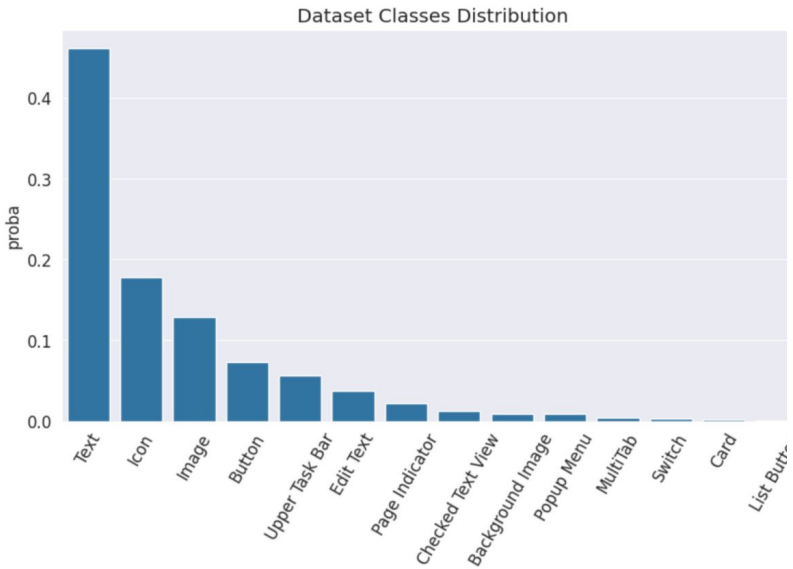
$$Q(R_m) = H(R_m) - \frac{|R_l|}{|R_m|}H(R_l) - \frac{|R_r|}{|R_m|}H(R_r)$$

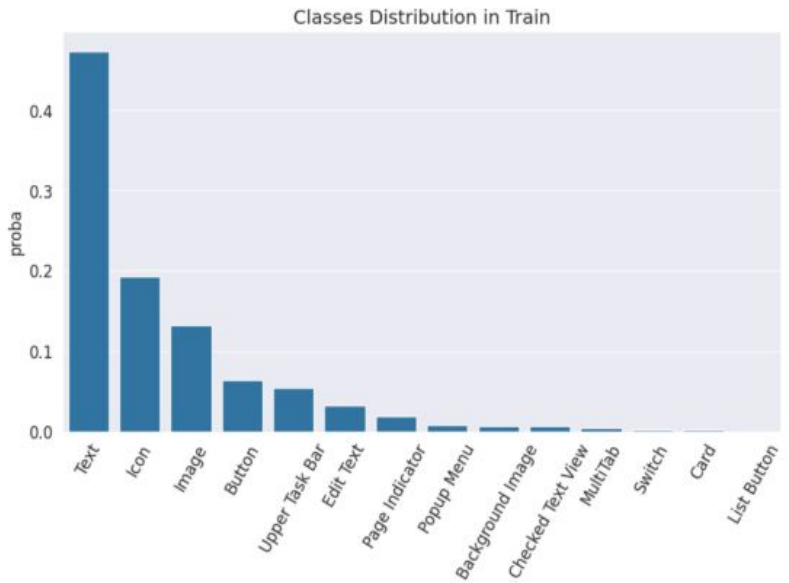
Функционал для decision tree (максимизируем его)

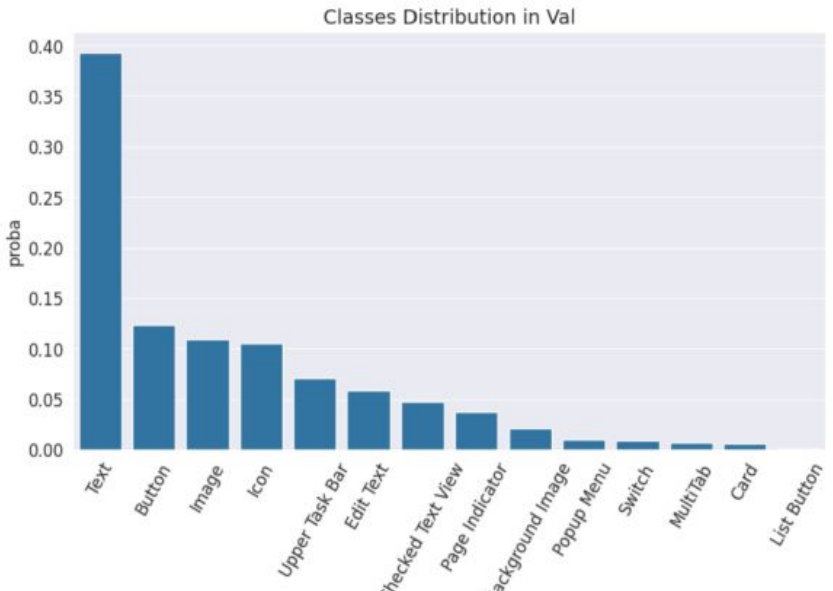
$$Q^*(R_l, R_r, \text{obj}) = \left| \frac{|R_l| \cdot H(R_l) + (|R_r| - 1) \cdot H(R_r) - (|R_l| - 1) \cdot H(R_l \setminus \{\text{obj}\}) - |R_r| \cdot H(R_r \cup obj)}{|R_l \cup R_r|} \right|$$

Лосс в нашем методе (минимизируем его при разбиении)

Как правильно делить выборку? Результаты







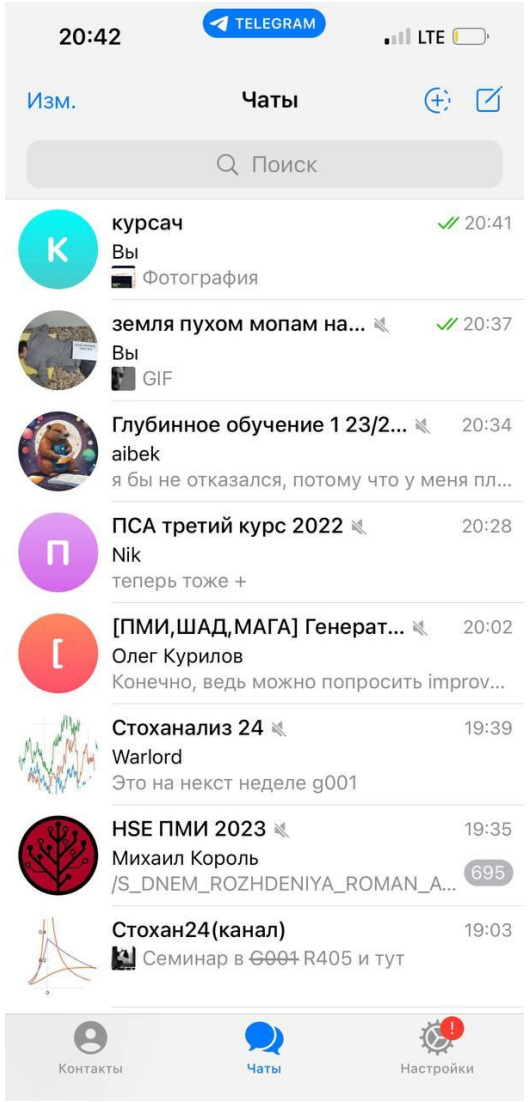
Результаты после fine-tuning'а моделей

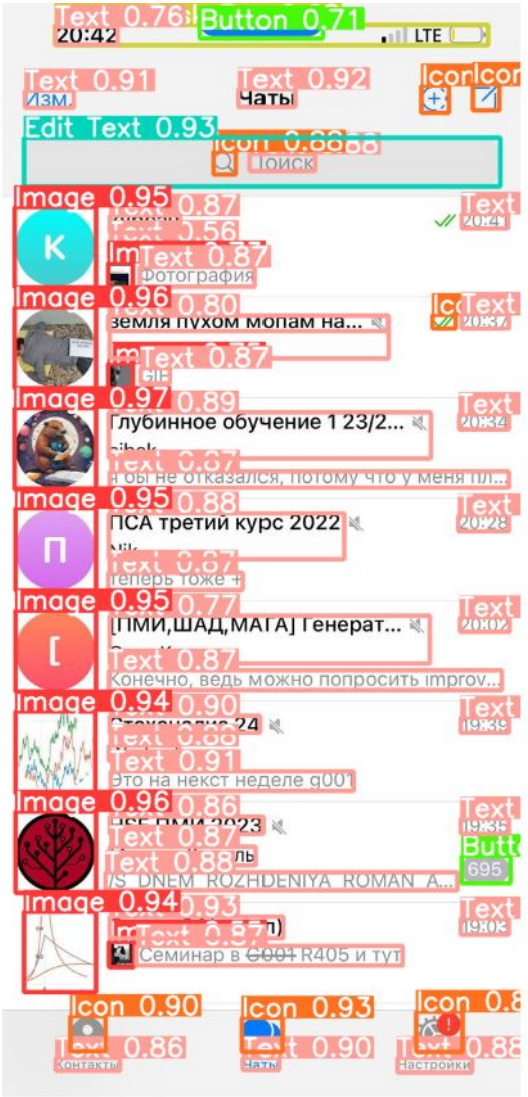
Model	mAP50	mAP@[50, 95]	# params (M)	Average Speed (ms)
YOLOv5	0.851	0.778	115.8	85.26
YOLOv8	0.866	0.824	**40.0**	68.14
YOLOv9	**0.878**	**0.835**	43.4	**57.39**

Все модели были предобучены на датасете COCO, каждая дообучалась на VINS Dataset 30 эпох

В итоге в качестве детектора выбрали YOLOv9.

Примеры разметки





Описание изображений

Задача: подготовка стабильной модели, способной описывать изображения разных доменов

Пример

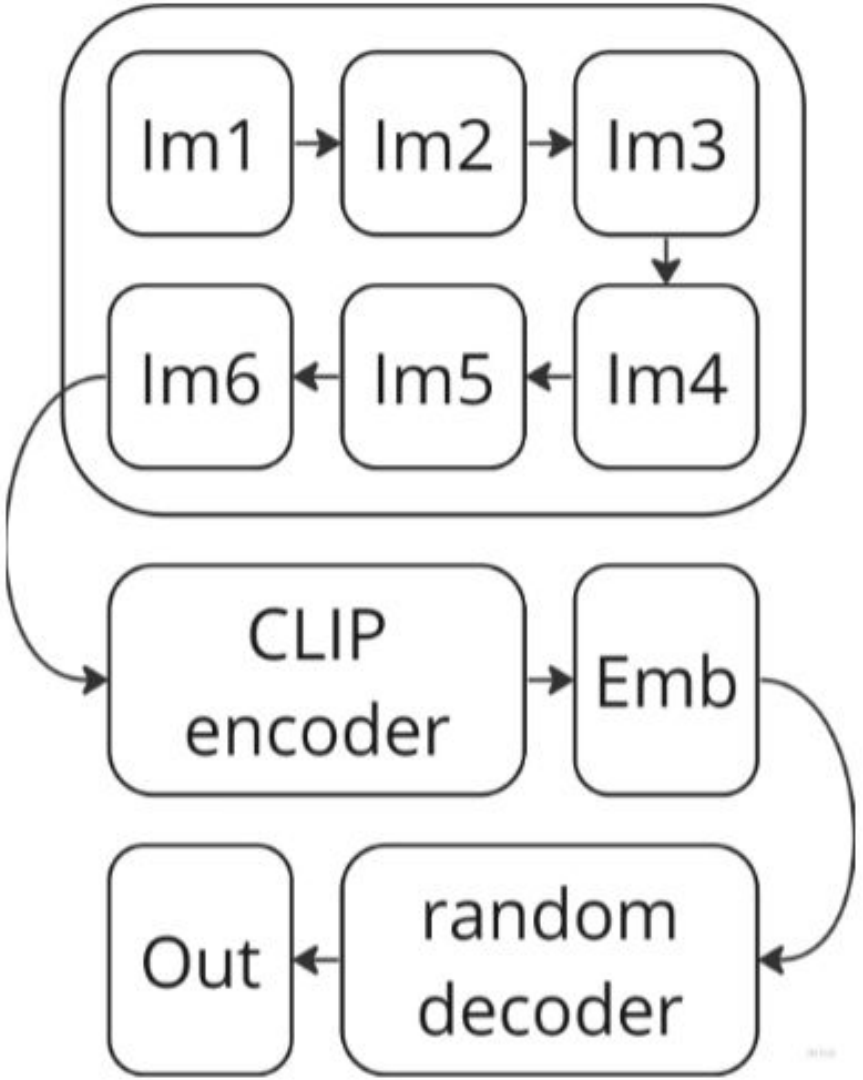
Caption: there is a red and white bus driving down the street.

Существующие решения

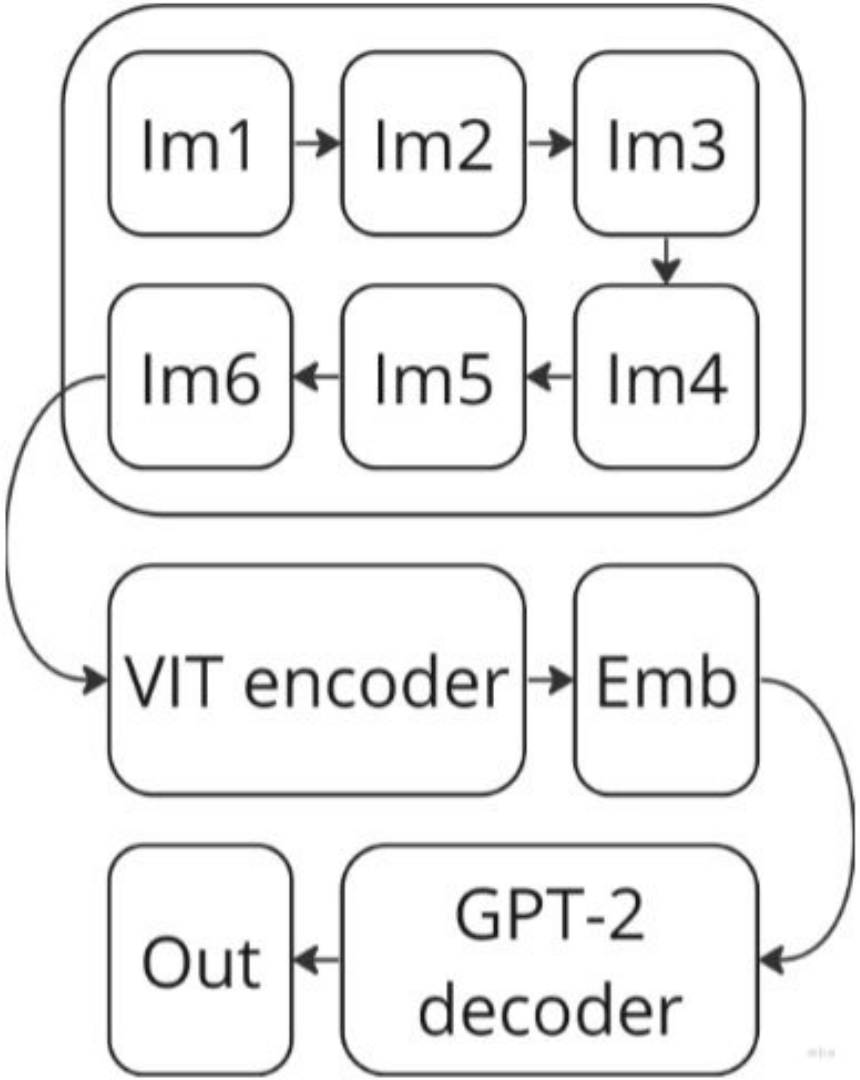
Описание изображений:

- GIT - модель состоящая из кодировщика CLIP и декодировщика GPT2
- VIT-GPT - модель состоящая из кодировщика VIT и декодировщика GPT2
- BLIP - трансформер (модель обучается одновременно на нескольких задачах, таких как описание изображений, вопросно-ответные системы и визуальная энтропия)

Существующие решения

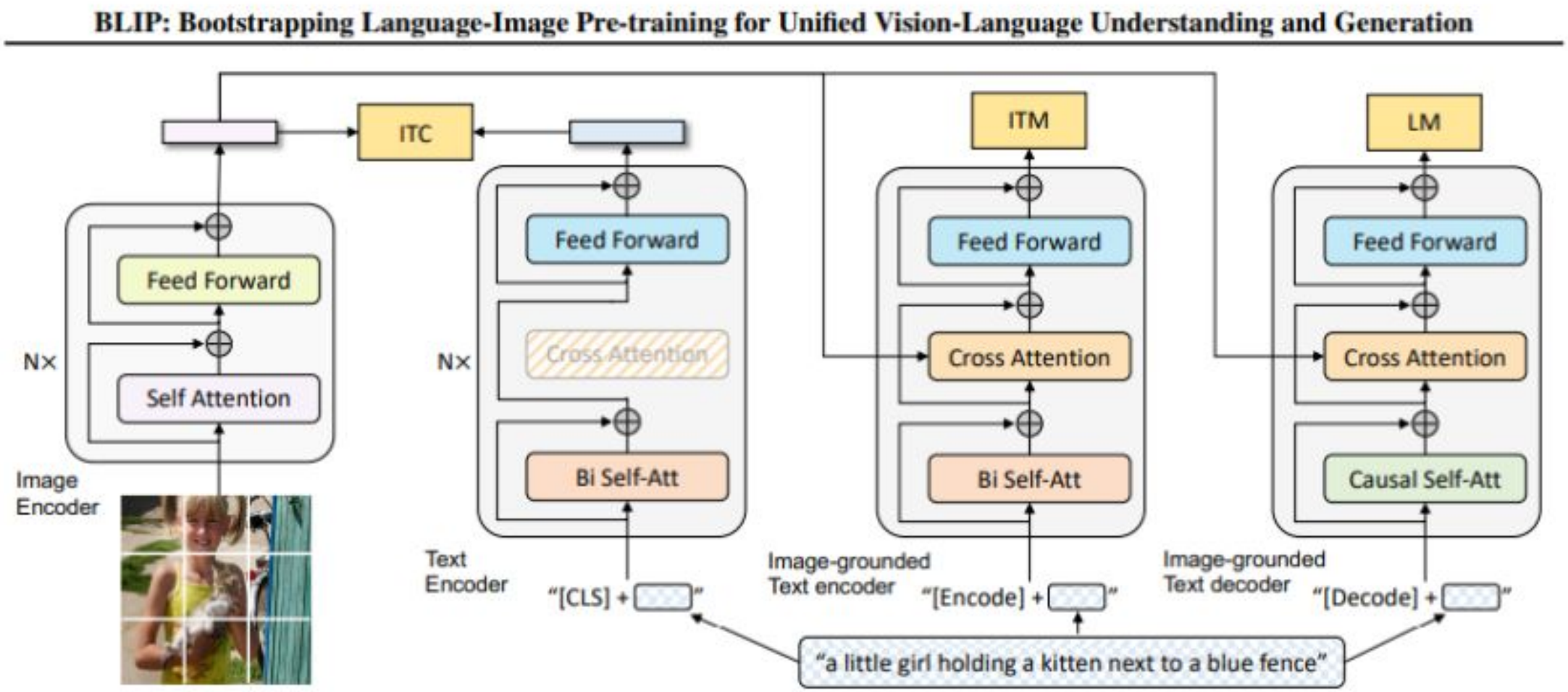


GIT



VIT-GPT

Существующие решения



BLIP

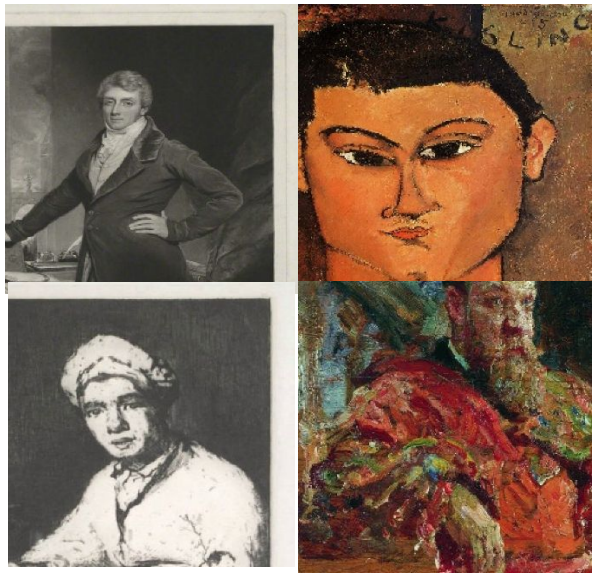
Датасеты

1) COCO - картинки смешанного типа
2) clothes - фотографии людей и их образов
3) cartoons - мультяшные и аниме изображения (104 произведения)
4) portraits - нарисованные портреты людей

COCO


clothes

cartoons


portraits

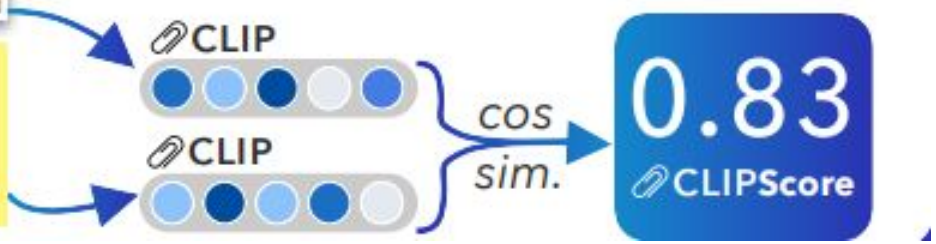
Метрики качества

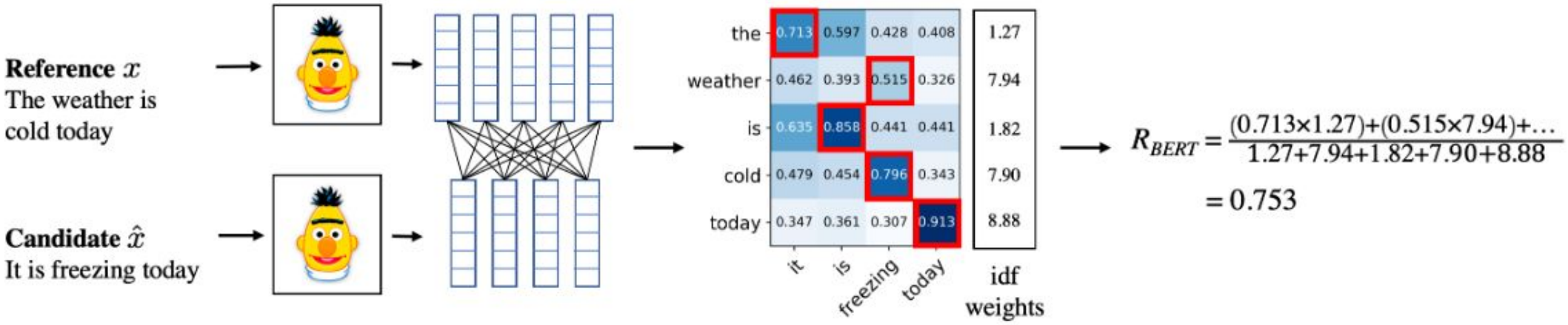
1) **CLIP score**
2) BERT score

REFERENCE CAPTIONS

- Two dogs are running toward each other across the sand.
- Two dogs run toward each other.
- Two dogs are running towards each other on a beach.

CANDIDATE
Two dogs run towards each other on a marshy area.





Сравнение моделей

Датасет: COCO
Метрика: CLIP score

Model	CLIPscore on COCO	Time (ms)
VIT-RuGPT	0.1831	755
GIT-Base	0.2715	183
BLIP-Base	0.2898	230
VIT-GPT2	0.2917	161
GIT-Large	0.2723	1513
GIT-Large-coco	0.3085	1490
BLIP-Large	0.3054	370

Результаты:

- Русская модель + базовые версии моделей не подходят по качеству
- VIT-GPT2 - не подходит по качеству
- GIT-Large - не подходит по времени
- **BLIP-Large - наиболее подходящая модель**

Сравнение моделей, примеры

Рис. 4.5

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. 4.6

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. 4.7

BLIP-Large: sonic the hedge running in a game.

BLIP-Base: sonic running through the jungle

GIT-Large: sonic the hedgehog game wallpapers

Рис. 4.8

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. 4.9

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. 4.10

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

Уязвимости BLIP

1. Возникают артефакты arafed, arafee и тд
2. Теряется информация о типе изображения (анимационное изображение/реалистичное изображение)
3. Уязвимость к полностью зашумленным изображениям
4. Неспособность адекватно описывать мемы

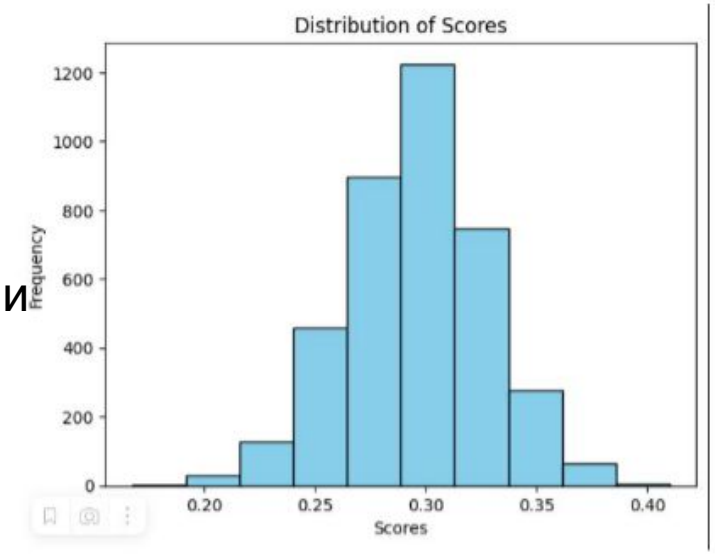


Рис. 4.3: Распределение CLIPscore на наборе данных COCO

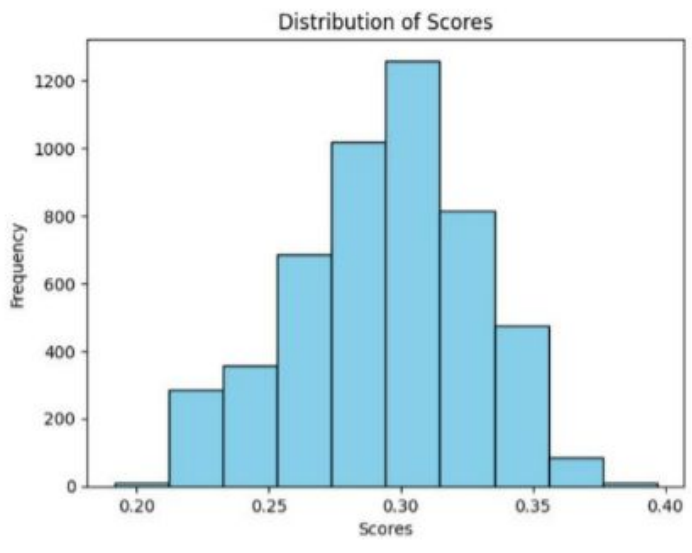


Рис. 4.4: Распределение CLIPscore на наборе данных cartoons

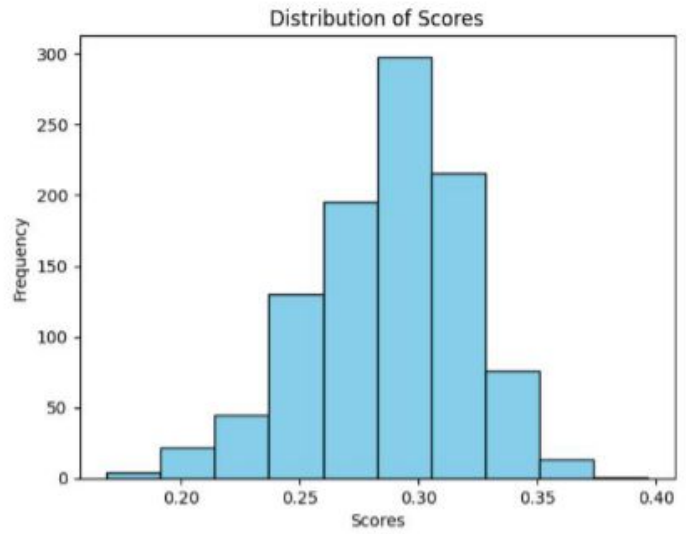


Рис. 4.11: Распределение CLIPscore на наборе данных "People Clothing Segmentation"

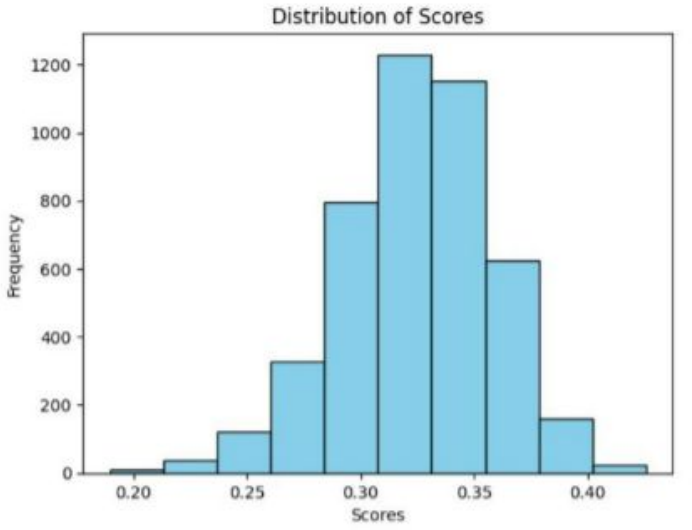


Рис. 4.12: Распределение CLIPscore на наборе данных "Portrait dataset for training GANs"

Уязвимость: arafed, arafee

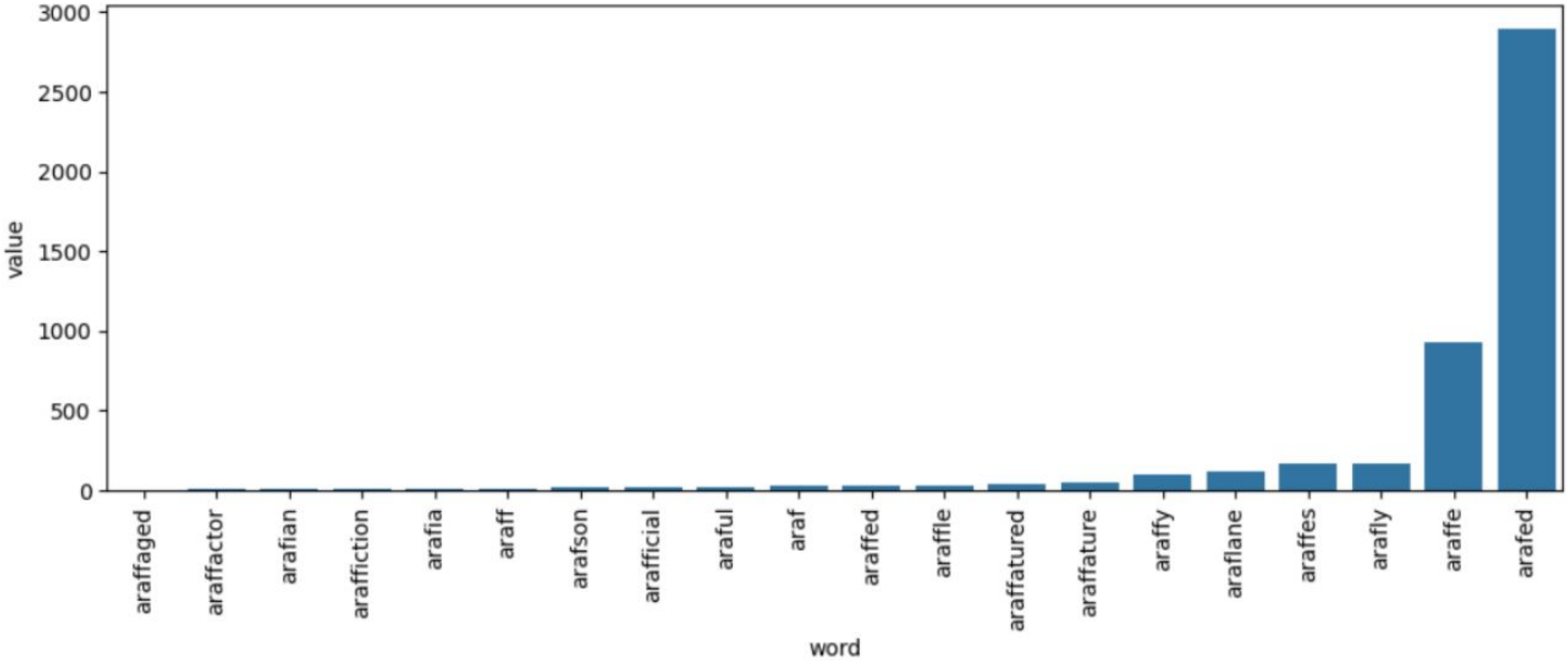


Рис. 4.13: Распределение артефактов araf*

Уязвимость: arafed, arafee

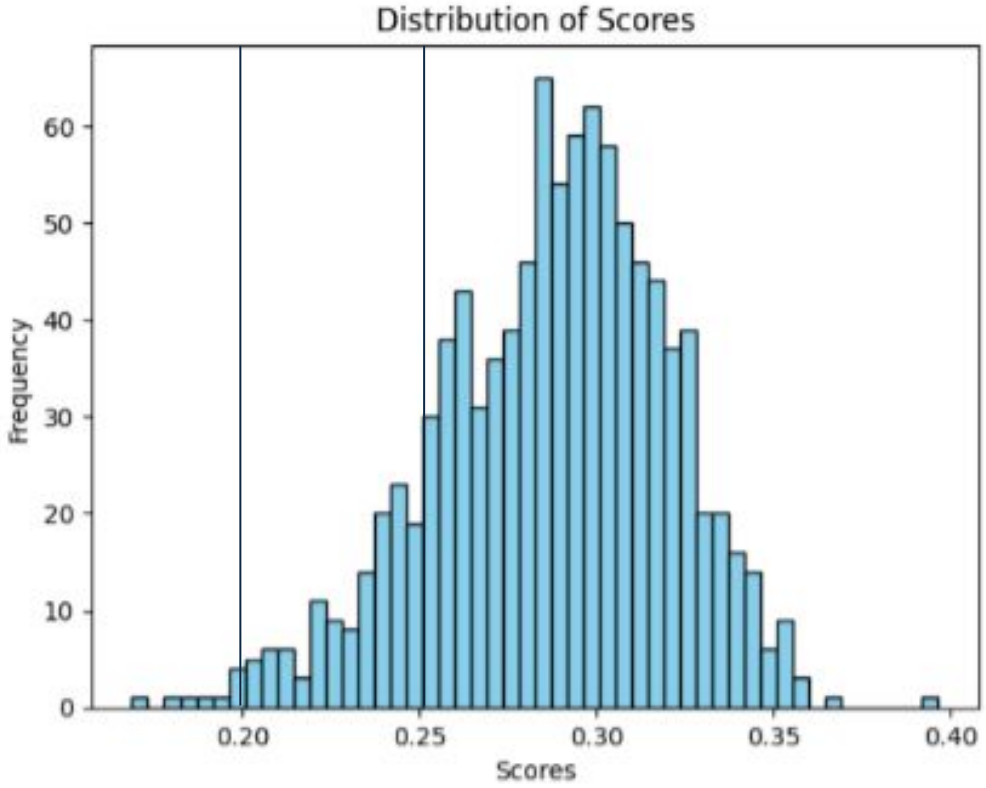


Рис. 4.14: До фильтрации

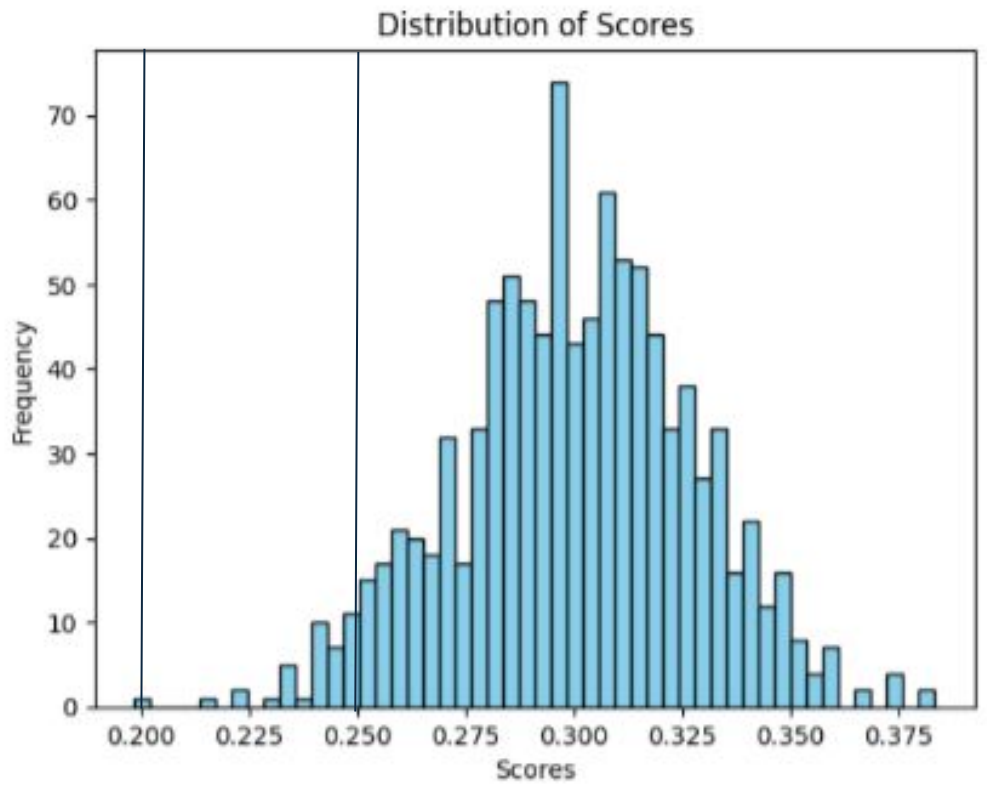


Рис. 4.15: После фильтрации

Проблемные предложения

- araffe on a boat with a mountain in the background
- araffe walking in front of a green door with a red umbrella

Решение: использовать префикс “An image of ”

Классификатор анимационных изображений

It’s a cartoon

CLIPscore: 20

It’s a photo

CLIPscore: 24

Photo

Классификатор анимационных изображений

Cartoons	Photos
It is an animation	It is a photo
It is a comic book	It is a photograph
It is a toon	It is a photo, it is a photograph
It is an animated series	It is a picture
It is a cartoon strip	It is an image
It is an anime	It is a snapshot
They are cartoon characters	It is a portrait
It is cartoony image	It is a print
It is a graphic novel	It is a photography
It is a cartoonish image	It is a visual photo
It is an anime image	It is a shot
It is a cartoon or anime image	It is a picturesque photo
Cartoon image	It is a photo not a cartoon image
It is a cartoon image, not a photo	It is a realistic photo
	It is a realistic image
	Realistic photo

Варианты разделения, предложенные LLM

Классификатор картинок

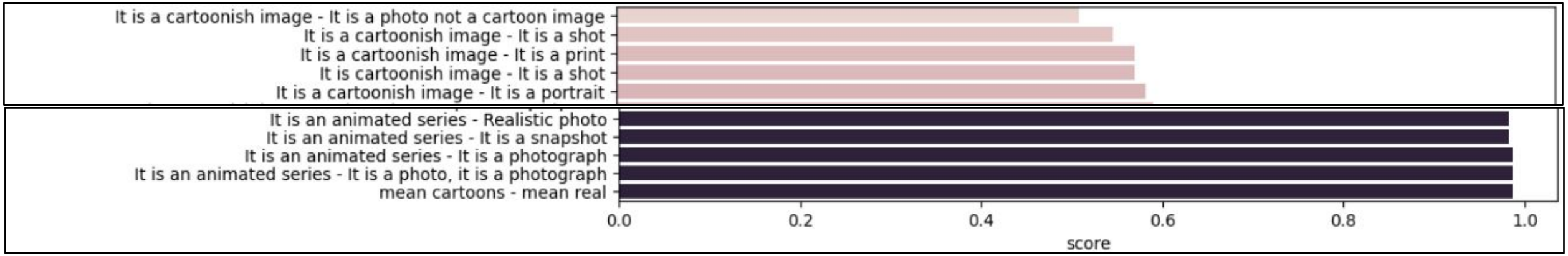


Таблица 4.3: Comparison of Model Performances and Sizes

Model	Image Phrase - Photo Phrase	Accuracy	Model Size
RN50x64	“It is an anime - It is a picturesque photo”	0.95944	1.26G
ViT-L/14	“It is an animation - It is a photo”	0.985332	890M
ViT-L/14@336px	“It is an animation - It is a photo”	0.989332	891M
RN50x4	“mean cartoons - mean real”	0.990555	402M
ViT-B/32	“mean cartoons - mean real”	0.990666	338M
RN101	“It is a cartoon or anime image - Realistic photo”	0.992888	278M
RN50x16	“It is an animation - It is a visual photo”	0.993444	630M
ViT-B/16	“It is a cartoon or anime image - It is a photo”	0.993444	335M
RN50	“It is a cartoon or anime image - It is a photo...”	0.994888	244M

Шум

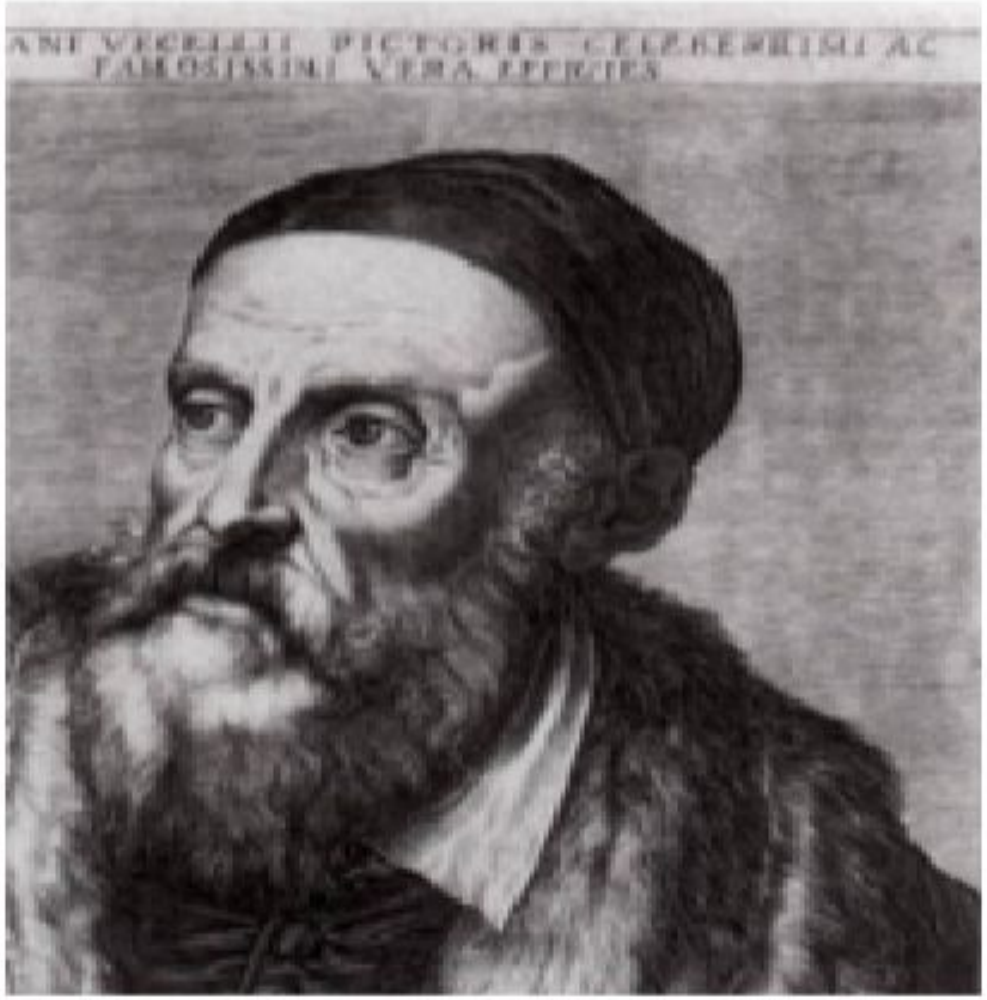


Рис. 4.17: Слегка шумное изображение

BLIP-Large: an old black and white photo of a man with a beard.

Рис. 4.18: Сильно зашумлённое изображение

BLIP-Large: there is a man, riding a skateboard on a ramp.

Шум

Рис. 4.19: Тёмное изображение

BLIP-Large: dimly lit room with a person standing in the dark.

Рис. 4.20: Осветлённое изображение

BLIP-Large: dimly lit room with a person standing in the dark.

Шум

Было решено построить распределения данной метрики на датасетах: cartoons, coco и noise (новый). Датасет noise содержит:

- Гауссовский шум
- Сплошной цвет (немного зашумлённый)
- Выборочные фотографии из датасета portraits, на которых ничего нет, кроме шума

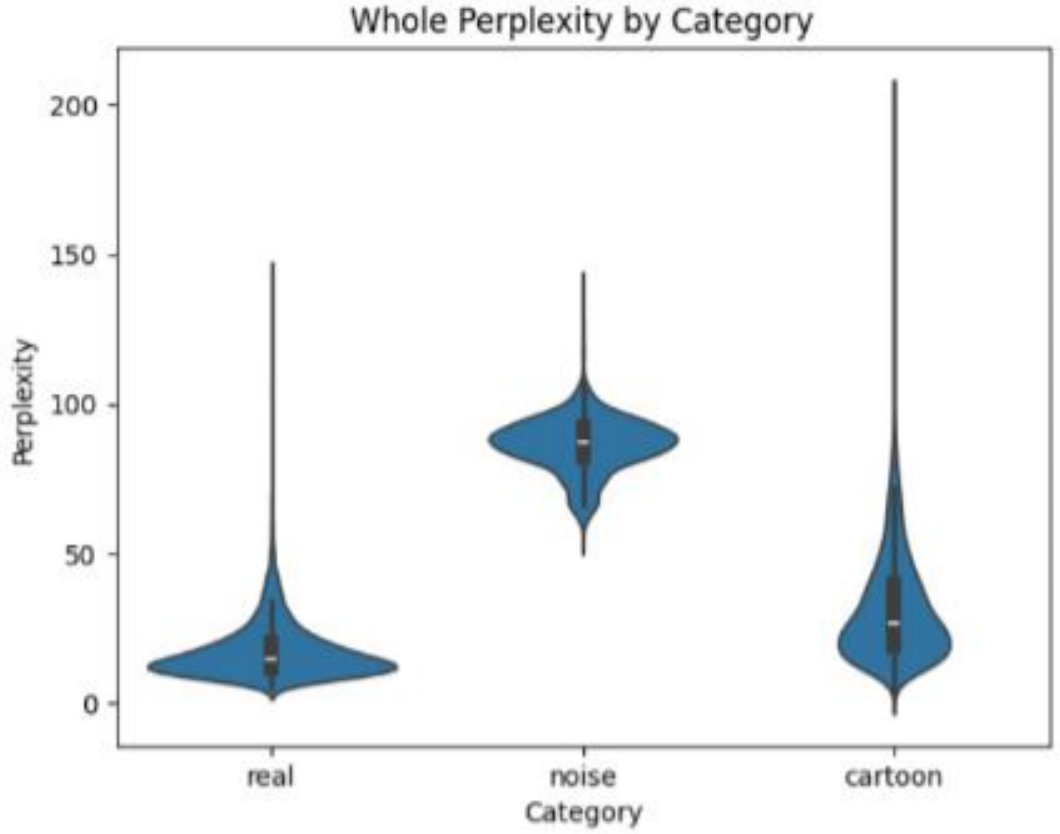


Рис. 4.21: Распределение уверенности модели

Модель детекции и распознавания текста

Задача : подготовка и развертывание моделей детекции и распознавания текста на картинке с фокусом на текст в естественной среде, например, уличные вывески и этикетки. Написание сервиса для взаимодействия с ними

ВИЖАЙ

Формальная постановка

- На вход модели детекции подается изображение
- Модель детекции выдает границы текстовой области в формате набора вершин четырехугольника а также уверенность предсказаний
- Вырезанные текстовые области подаются на вход модели распознавания
- Модель распознавания возвращает текст и уверенность в предсказании

Детекция: обзор существующих методов

Регрессионный подход : **EAST**

Разделение близко расположенных текстов :
PSENet

Детекция текста со сложной геометрией : **FCENet**

Трудный пост-процессинг в сегментационных моделях:
DBNet
Предсказывает различные геометрические
свойства текстового многоугольника и комбинирует их :
SAST

Обработка текста различного масштаба без потери качества : **DBNet++**

Распознавание: обзор

Использует сверточные слои для обработки изображения и BiLSTM
для формирования текста : **CRNN**

Использует единую визуальную компоненту, улавливает внутрисимвольные
различия а также взаимосвязи между символами и контекст их расположения : **SVTR**

Интегрирует лингвистические знания, итеративно улучшая предсказание : **ABINet**

Добавляет модули коррекции хроматических искажений и искажений формы : **SPIN**

Улучшает стандартную encoder - decoder архитектуру вводя модуль с усиленной
позиционной информацией : **Robust Scanner**

Метрики детекции

- **Asis**
- **Join**
- **TedEval**

$$\left[IoU(\text{bounding box, ground truth}) > 0.5 \right.$$
$$\left\{ \begin{array}{l} IoU(\text{bounding box, ground truth}) > 0.2 \\ \dfrac{\text{area of overlap (bounding box, ground truth)}}{\text{area (bounding box)}} > 0.9 \end{array} \right.$$

Условия сопоставления рамок в **asis** алгоритме

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

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

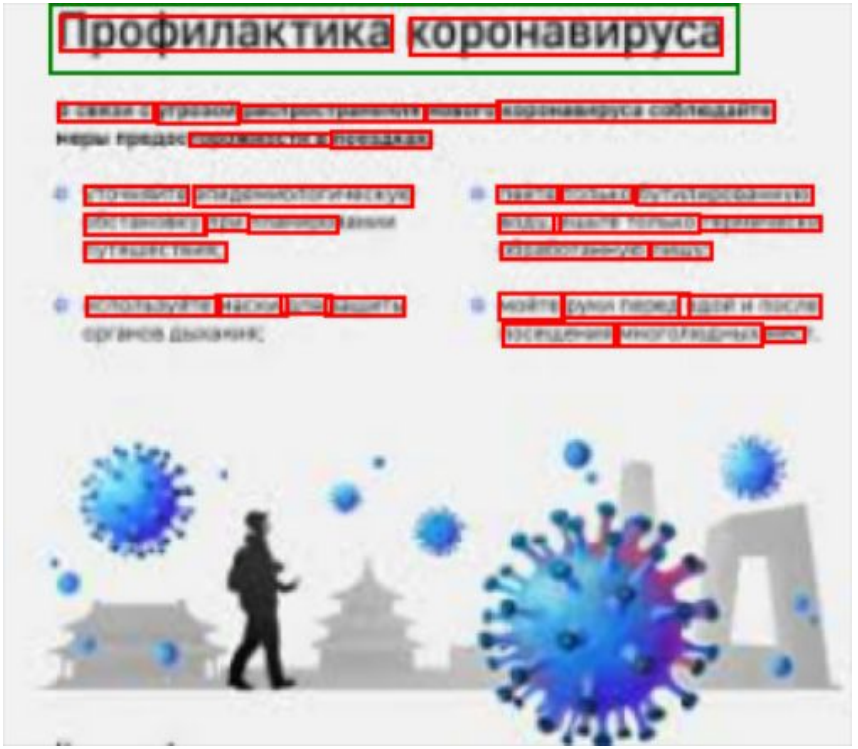
Метрики распознавания

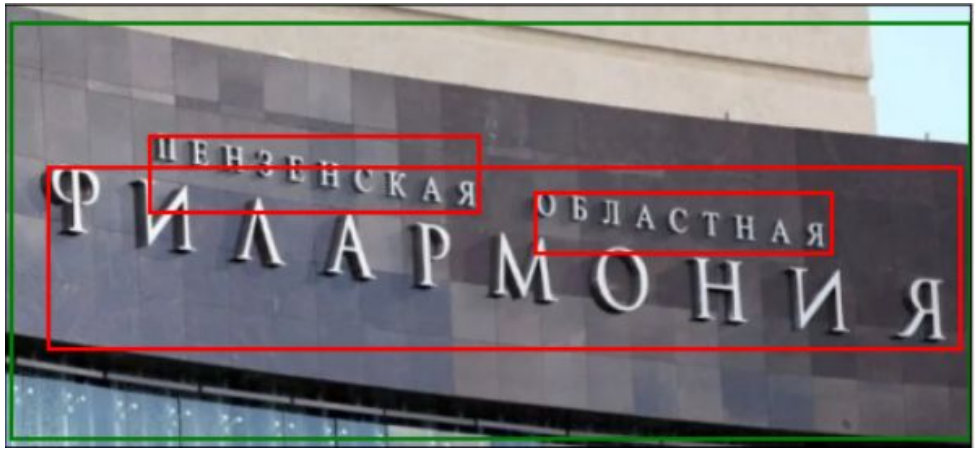
- **Точность**
- **Normalized Edit Distance**

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

Выбор модели детекции

DBNet++





Выбор модели распознавания

- **Синтетический датасет** - вырезанные текстовых частей из синтетических изображений, предназначенных для детекции.
- **Реальный датасет** - вырезанные задетектированные с большой уверенностью области реальных изображений

pred_text = Далу real_text = Lazy

pred_text = веаиту real_text = BEAUTY

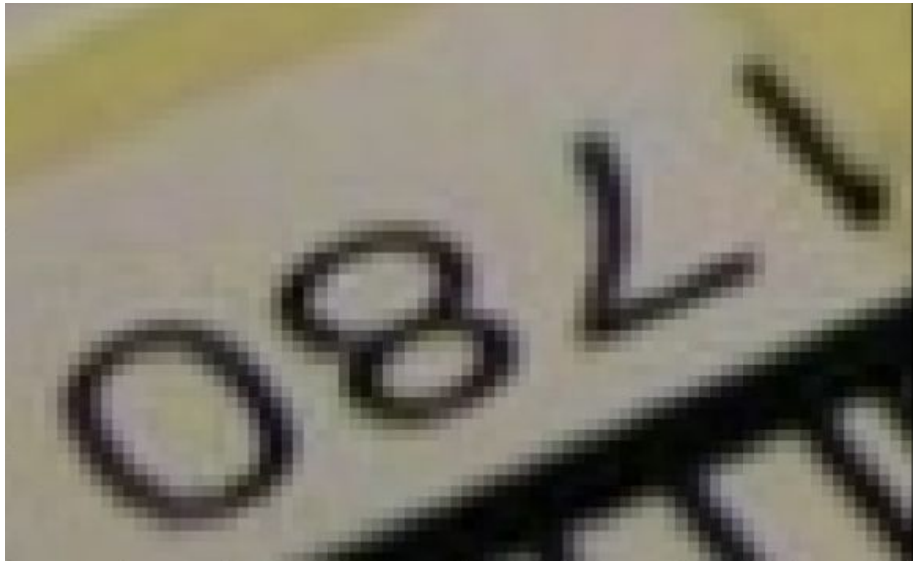
Пример распознавания **SVTR**

Проблема смещения синтетических данных

- Включение реальных изображений в обучение
- Создание новых словарей : advance (русский+английский+цифры+спецсимволы), light (без спецсимволов)
- Дообучение SVTR на синтетике и реальных изображениях с advance словарем

pred_text = bcerga. real_text = всегда

pred_text = 082! real_text = 1780



Проблема балансировки данных

- Было замечено что модель достаточно часто пустала английские и русские буквы
- Был выявлен дисбаланс в распределении английского текста между реальными и синтетическими данными
- Был проведен следующий эксперимент: в процессе обучения доля английского и русского текста поддерживалась 1:1.

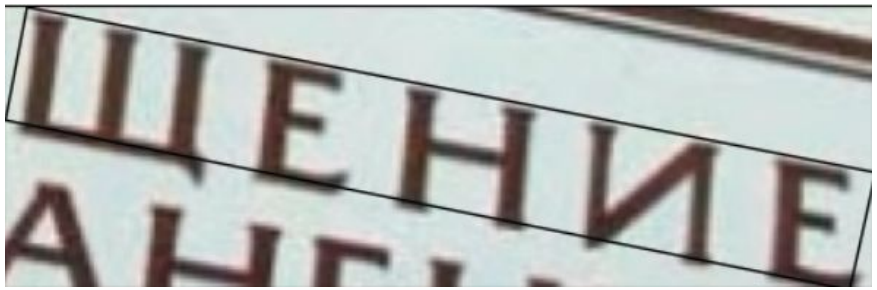
Примеры предсказаний модели где присутствует проблема с распознаванием английских символов

Эксперименты с разными словарями

- Дообучение модели на словаре advance, исключив из него заглавные буквы.
- Точность модели улучшилась в плане распознавания буквенных символов
- Модель стала лучше справляться с распознаванием отсутствия текста на изображении
- Улучшилась ситуация с перепутыванием английских и русских символов.
- Эксперимент с использованием словаря light
- Анализ результатов подтвердил, что модель достаточно часто ошибочно принимает спецсимволы за другие символы

Эксперименты с данными

- Обработка многострочного повернутого текста: техника **экстраполяции**
- Обработка повернутого текста: **корректировка угла поворота**

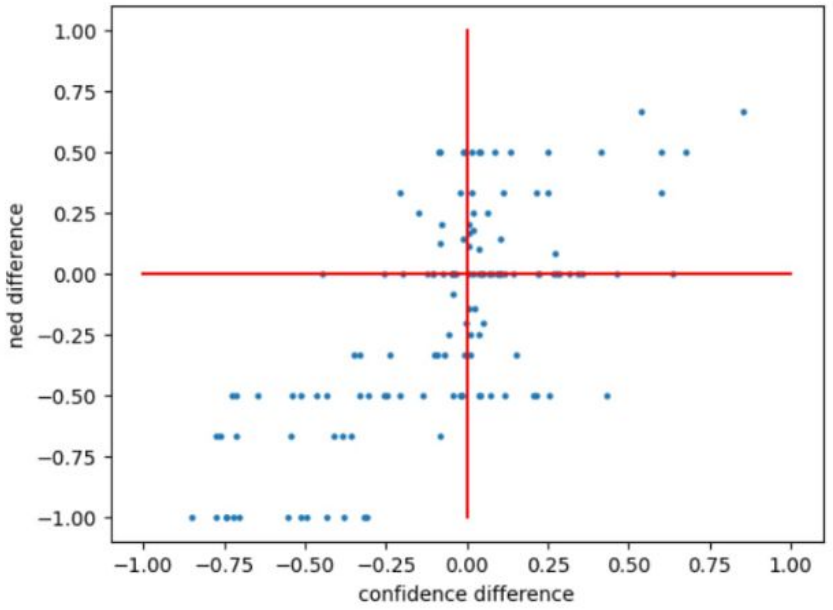


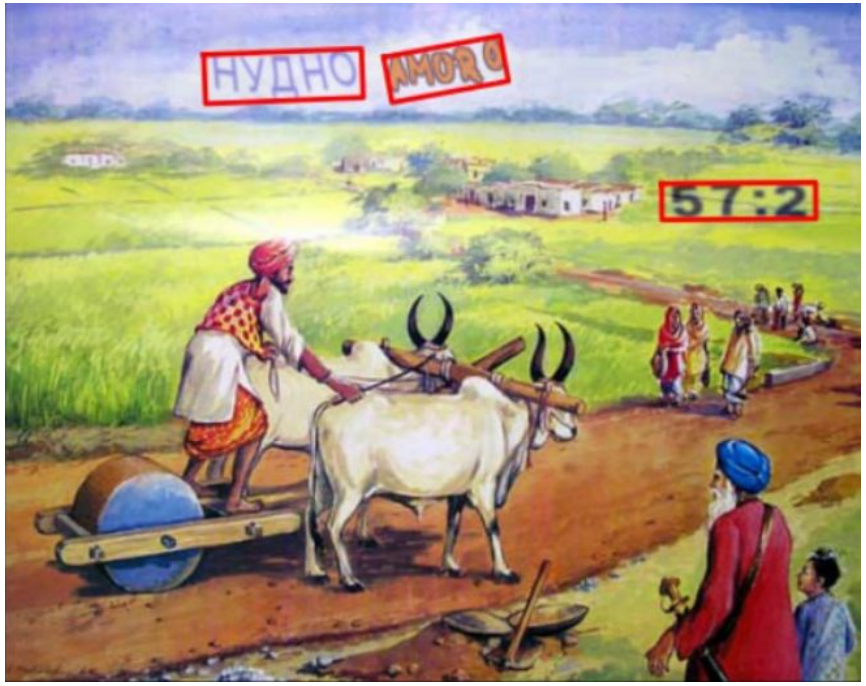
Эксперименты с улучшением синтетического набора данных

Был построен собственный пайплайн для генерации синтетических данных, основанный на статистиках, полученных по реальным данным

Модель обученная с коррекцией поворотов менее устойчива

Использование коррекции поворота только во время инференса и выбирать предсказанный текст основываясь на уверенности





Эксперименты со смешиванием разных видов синтетики

Балансировка доли реальных данных и новой синтетики в процессе обучения

Добавление данных из старой синтетической выборки в обучение

Старая синтетика покрывает часто встречающиеся паттерны, в то время как новая синтетика покрывает как можно больше сложных случаев

В начале модель обучается на смеси старой синтетики и реальных данных, после этого она дообучается на новой синтетике в сочетании с реальной выборкой

Добавление пробельного символа

Определение возраста и пола человека по изображению

Задача: подготовка и развертывание моделей детекции человека (включая лицо, туловище) и определения возраста и пола по полученному изображению, Написание сервиса для взаимодействия с ними

→ **36 years old woman**

Обзор наборов данных:

IMDB+Wiki

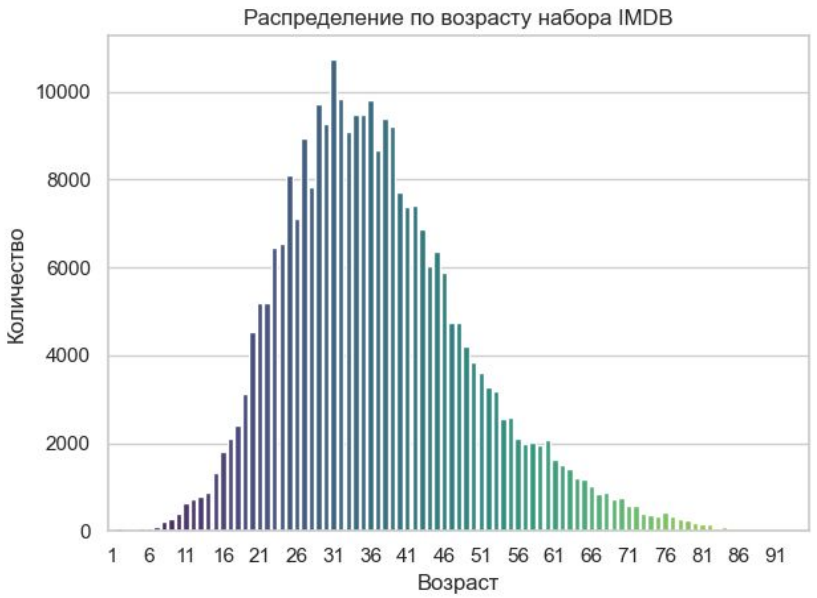
(cleaned – 286 000 изображений)

IMDb

460,723 images

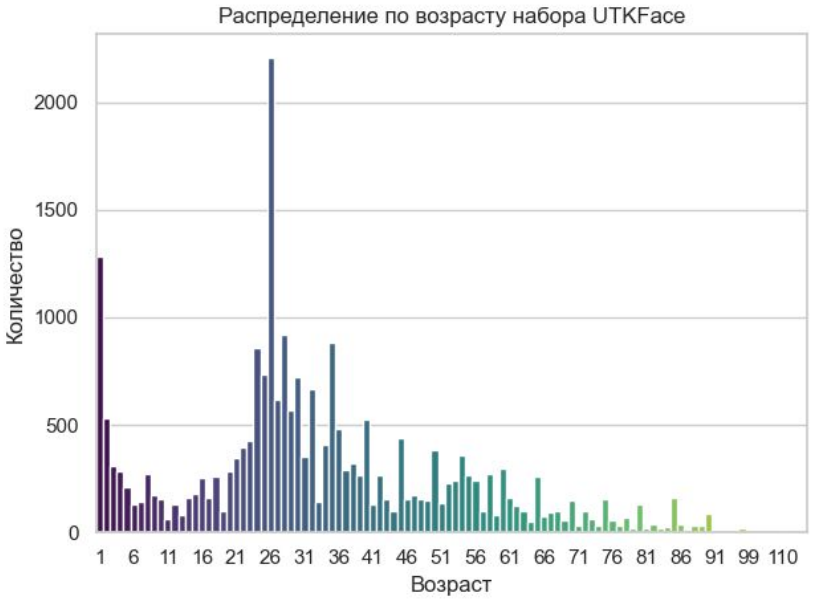
Wikipedia

62,328 images



UTKFace

(20 000 изображений)



Детекция: существующие методы

- **Haar Cascade** - основан на признаках Хаара, которые представляют собой специфические структуры яркости на изображении для обнаружения лиц.
- **MTCNN** (Multi-task Cascaded Convolutional Networks) - модель, использующая каскад из трех сверточных нейронных сетей для обнаружения лиц и одновременного определения их границ и ключевых точек.
- **YOLO v5** (You Only Look Once) - это архитектура детектора по разпознаванию объектов в реальном времени. YOLO состоит из двух частей: encoder (свёрточные слои) и head (классификационный слой).
- **YOLO v8** - архитектура, представляющая собой более актуальный вариант в серии YOLO.

Детекция: сравнение методов

Для сравнения моделей детекции объектов используется метрика Intersection over Union (IoU) 1. При значении IoU между предсказанными координатами и реальными аннотациями не менее 0.5 объект считается обнаруженным.

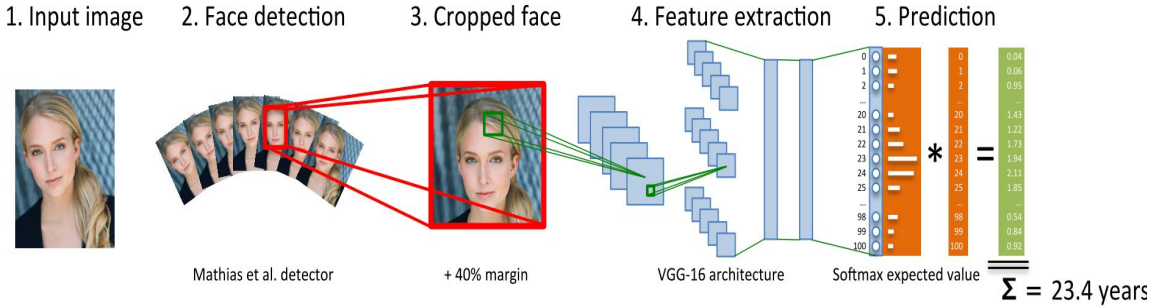
Таблица 4.5: Сопоставление предобученных моделей детекции лица на тестовой подвыборке IMDB-clean (56087 изображений).

Model	Faces detected	Labels missed	#params(M)	Average time (ms)
Haar Cascade	179664	2992	-	38.2
MTCNN	128431	105	2.31	155.6
YOLO v5	184318	0	1.73	40.94
YOLO v8	178415	23	3.01	13.52

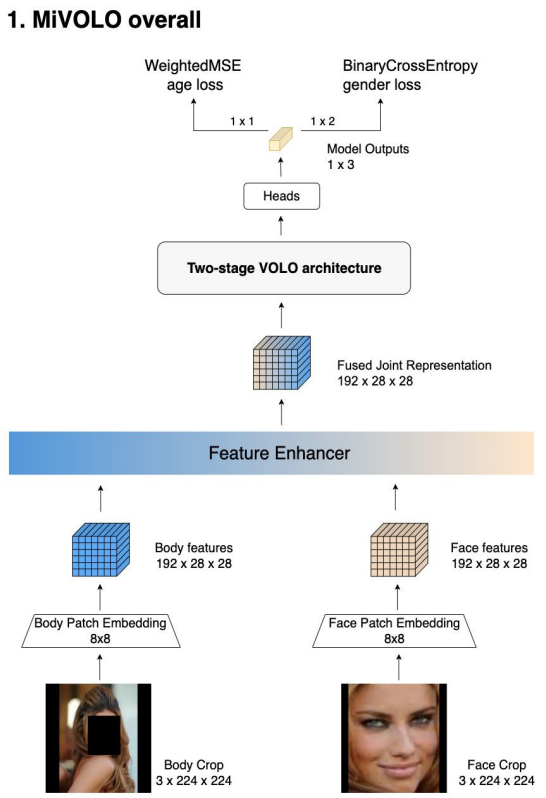
В дальнейшем применялась модель детекции YOLO v8

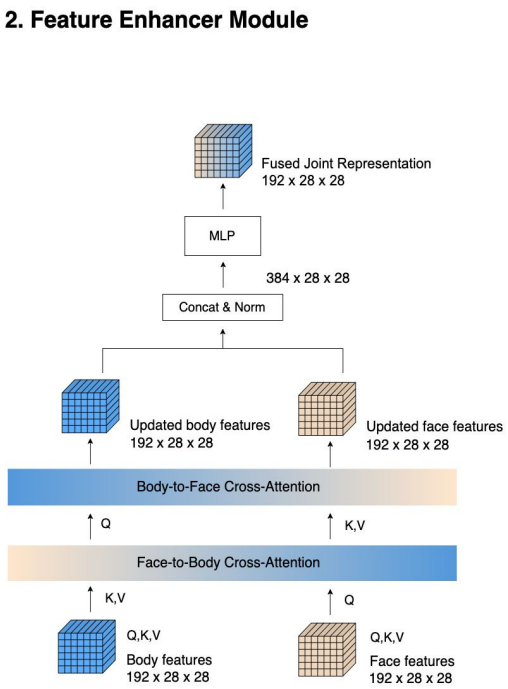
Модели определения возраста и пола: существующие методы

DEX (Deep EXpectation)



MiVOLO (Multi-input Transformer for Age and Gender Estimation) - SOTA-модель





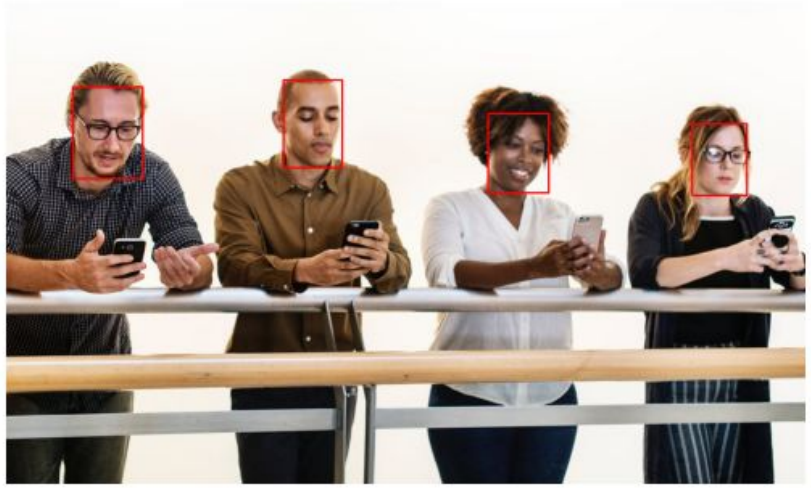
Метрики:

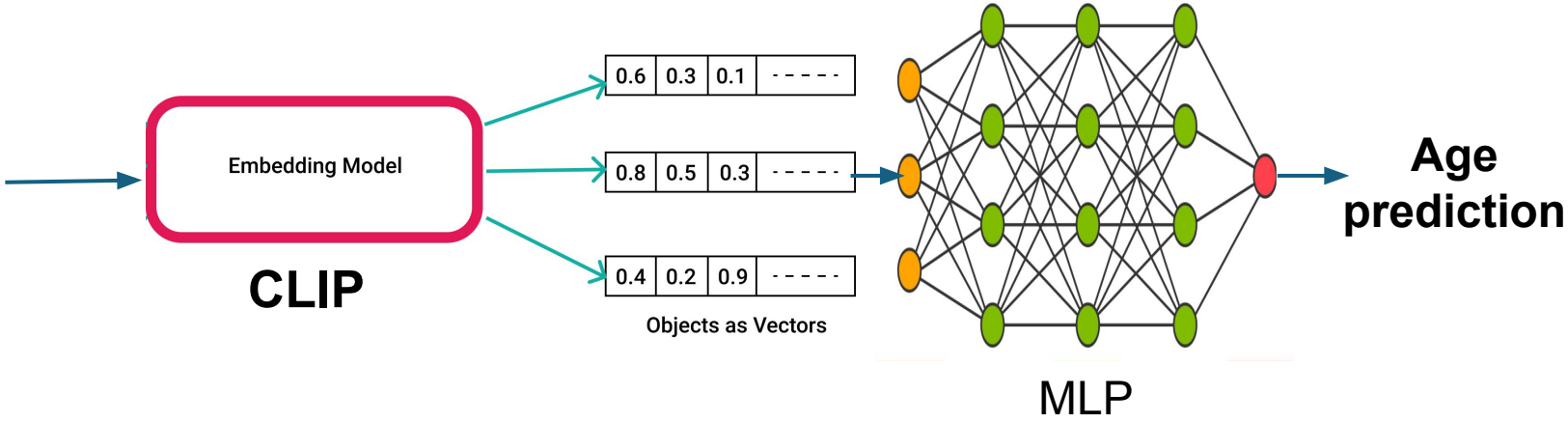
$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$\mathrm{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Модель определения возраста и пола: собственная реализация с использованием CLIP

Предлагаемый подход включает в себя использование предобученной модели **CLIP** (Contrastive Language–Image Pre-training) для извлечения векторных представлений из изображений. После получения эмбеддингов использовался MLP для решения регрессионной задачи.





В данной реализации относительно небольшой MLP, в котором несколько линейных слоев и функций активации.

Модель определения возраста и пола: сравнение моделей

Таблица 4.8: Результаты CLIP-based модели.

Train Dataset	Test Datset	Age (MAE)	Average time (ms)
UTKFace	UTKFace	4.2	11.06
UTKFace	IMDB-clean	14.23	15.17

Таблица 4.6: Сопоставление предобученных моделей на тестовой подвыборке IMDB-clean (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 (face-only)	6.46	99.3	25.86	5.63
MiVOLO (face-and-body)	5.5	99.52	27.43	10.3

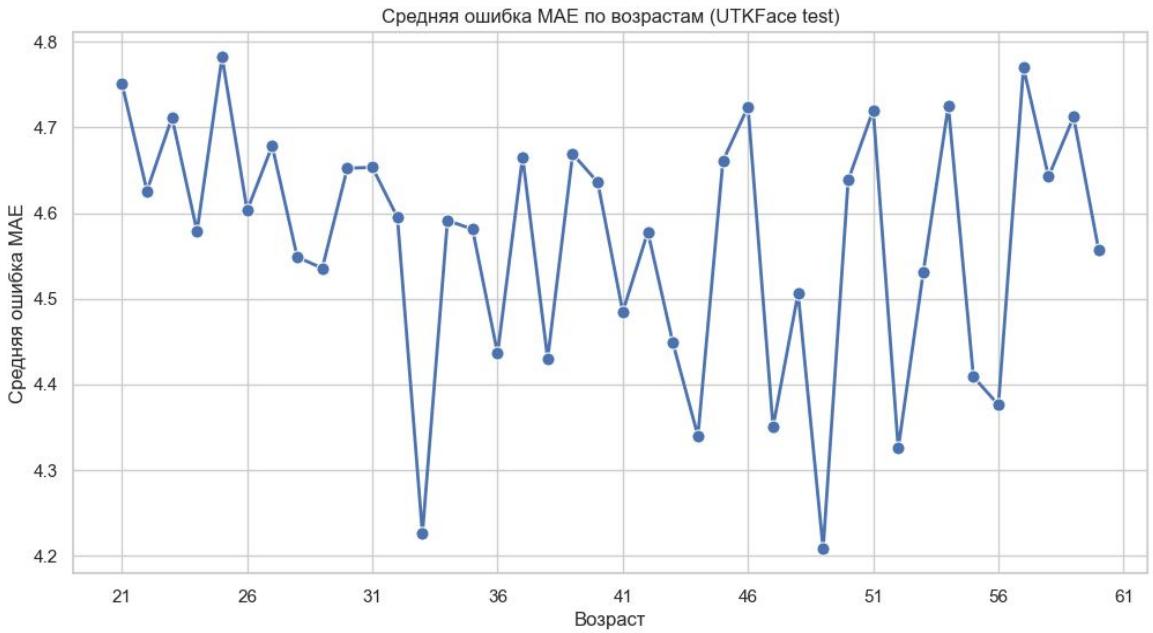
Таблица 4.7: Сопоставление предобученных моделей на тестовой подвыборке UTKFace (3553 изображений).

Model	Age (MAE)	Gender (Acc)	#params (M)	Average time (ms)
DEX	7.16	78.9	134.7 + 134.2	7.4 + 5.93
MiVOLO (face-only)	5.02	97.3	25.86	3.28
MiVOLO (face-and-body)	4.6	97.63	27.43	6.2

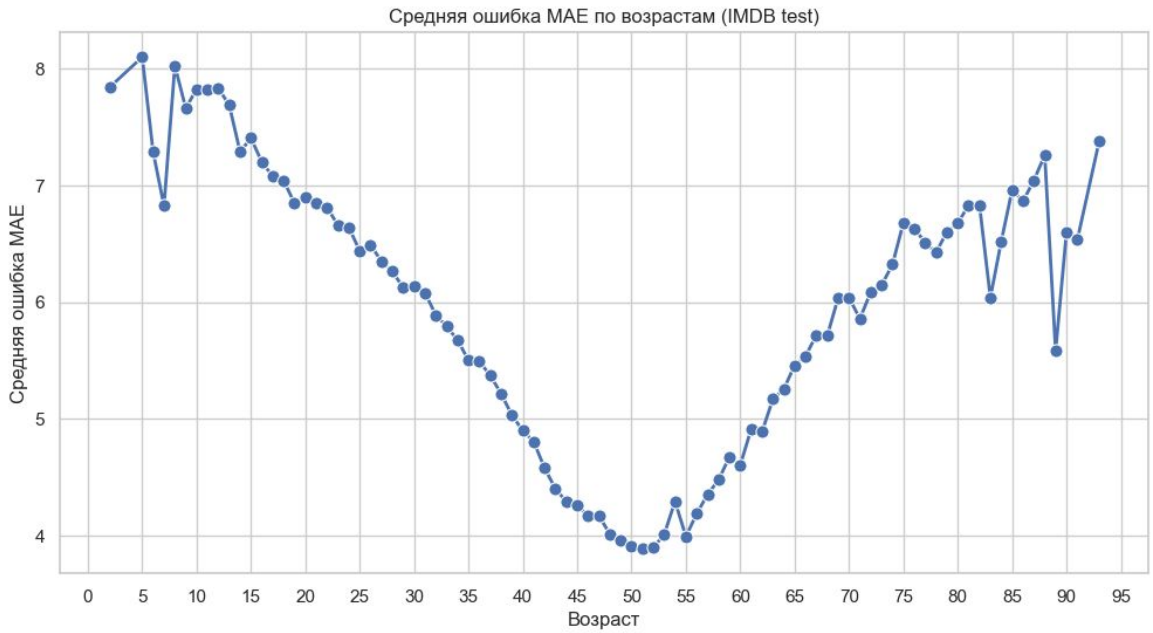
Таким образом, вместо усложнения CLIP-based архитектуры или DEX, были сосредоточены усилия на улучшении и оптимизации модели MiVOLO.

Модель определения возраста и пола: MiVOLO

Главный недостаток - низкое качество на хвостах распределения



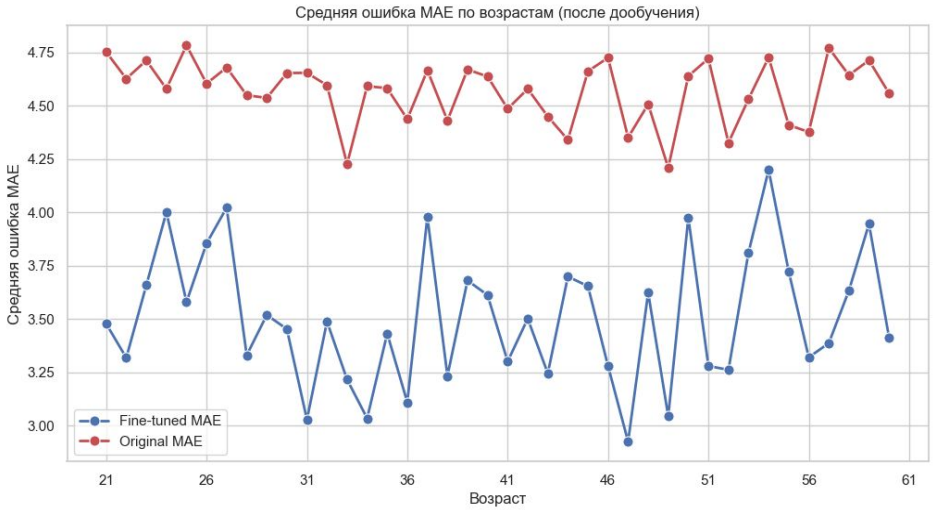
Тестовая подбвыборка (3553 изображений)



Тестовая подбвыборка (56087 изображений)

Модель определения возраста и пола: MiVOLO. Результат.

Дообучение производилось на расширенном наборе данных, полученном путем слияния подвыборок IMDB-clean и UTKFace, с особым упором на крайние возрастные группы.



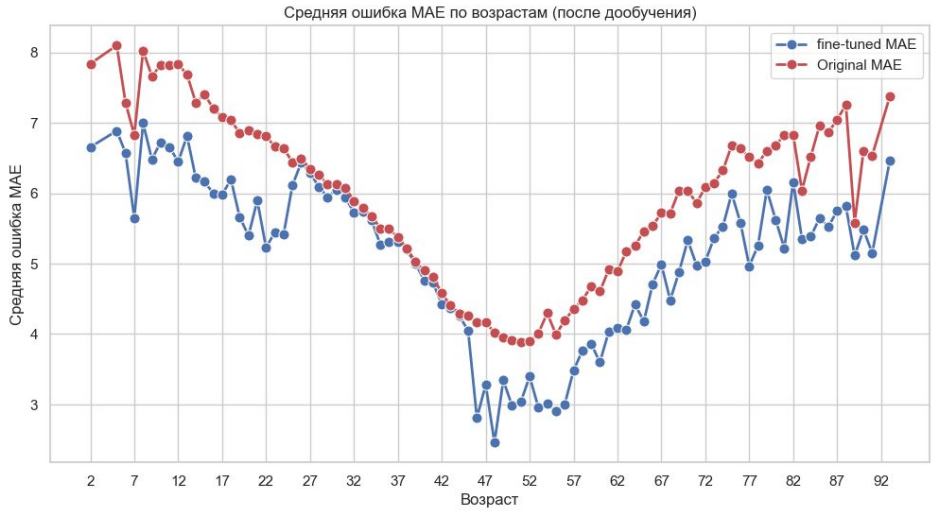


Таблица 4.9: Результаты модели MiVOLO (face-and-body) на различных тестовых выборках.

Test Set	Age (MAE)	Gender (Acc)	#params (M)	Average time (ms)
UTKFace	3.77	97.52	27.43	6.2
IMDB-clean	4.69	99.46	27.43	10.5

Модель детекции и определения возраста и пола: Заключение.

- Выбранная модель детекции - **YOLO v8**
  - Пороги модели - по умолчанию модели, так как таким образом получается лучший результат.
- Выбранная модель определения возраста и пола - **MiVOLO**
  - Средняя уверенность модели при неверных предсказаниях составляет 0.924 на наборе данных UTKFace, для верных предсказаний - 0.978. Поэтому установлен порог - 0.94. В случаях, когда уверенность модели ниже этого порога, идентифицированный объект классифицируется как "person" что позволяет избегать ошибочного определения пола при низкой уверенности предсказаний
  - Сам конечный ответ выглядит, как "[age] years old man / woman / boy / girl / person где [age] — это округлённый возраст.

Реализация сервиса

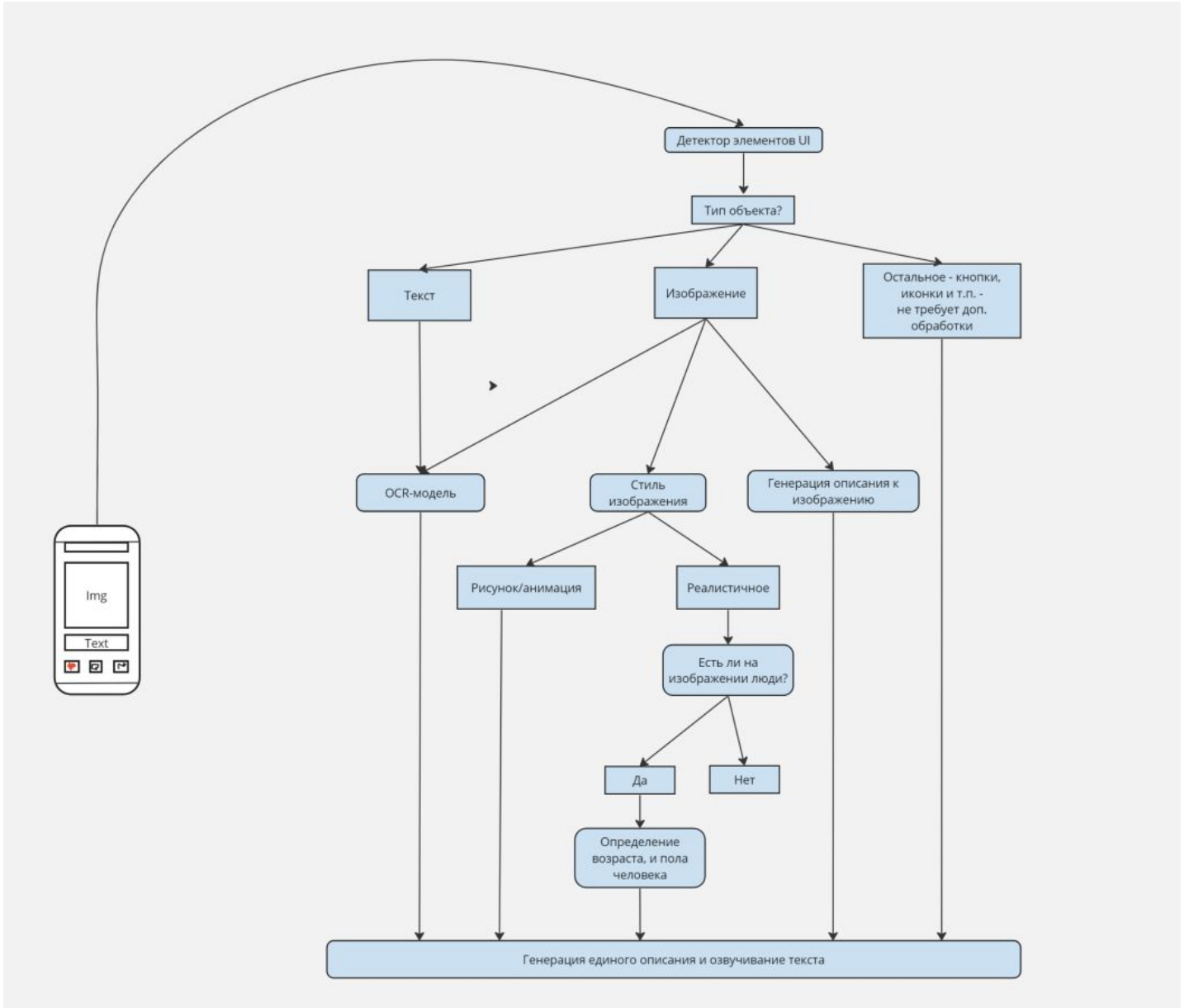
Использовался **Docker**, чтобы каждая модель запускалась в своём окружении, не мешая остальным

Каждый контейнер включает в себя сервис, написанный с использованием **FastAPI**

Реализовали передачу информации с помощью настройки сети между контейнерами с использованием инструмента **Docker compose**.

Реализовали взаимодействие с нашим сервисом через асинхронного **телеграм-бота.**

Реализация сервиса



Эвристики

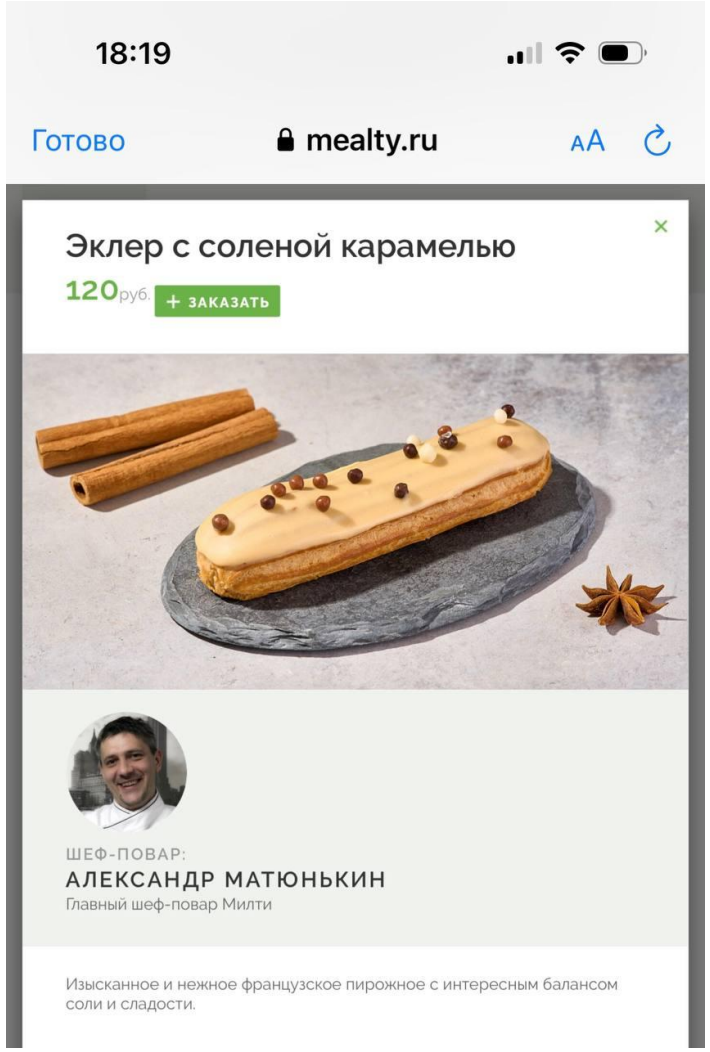
- **Детекция UI:**
  - Разбиение изображения на 9 пространственных частей, каждой из которых присваиваем свои объекты

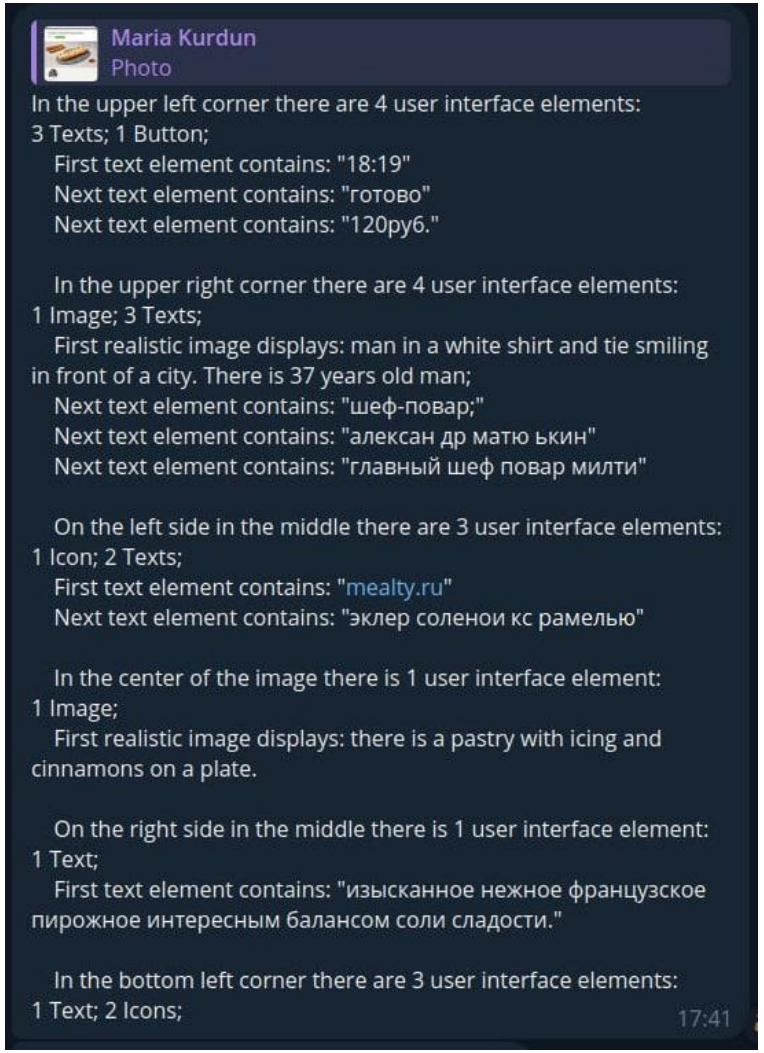
- **Определение возраста и пола человека по изображению:**
  - Не анимационные объекты, идентифицированные на изображении, перечисляются в порядке убывания площади их рамок детекции.

- **Детекция и распознавание текста:**
  - Текстовые элементы обрабатываются и с использованием модели детекции, и без нее, итоговый текст выбирается по уверенности модели

  - Задетектированные текстовые части сортируются внутри изображений и текстовых элементов приближенно к порядку чтения

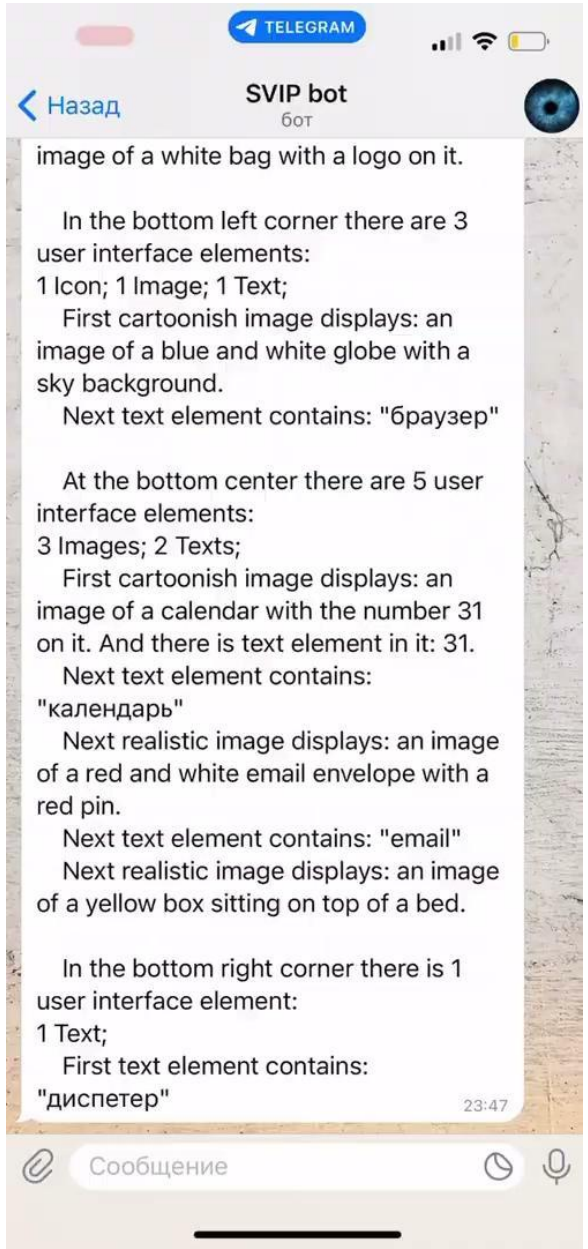
Результат





+Аудио сообщение с озвучиванием текста

+JSON с метаданными


TELEGRAM
Назад
SVIP bot
бот
image of a white bag with a logo on it.
In the bottom left corner there are 3 user interface elements:
1 Icon; 1 Image; 1 Text;
First cartoonish image displays: an image of a blue and white globe with a sky background.
Next text element contains: "браузер"
At the bottom center there are 5 user interface elements:
3 Images; 2 Texts;
First cartoonish image displays: an image of a calendar with the number 31 on it. And there is text element in it: 31.
Next text element contains: "календарь"
Next realistic image displays: an image of a red and white email envelope with a red pin.
Next text element contains: "email"
Next realistic image displays: an image of a yellow box sitting on top of a bed.
In the bottom right corner there is 1 user interface element:
1 Text;
First text element contains: "диспетер"
23:47
Сообщение

Вопросы