

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:
Программное обеспечение для определения цвета зубов/прикуса

Выполнили студенты:

группы #БПМИ224, 2 курса

Вдовец Илья Давидович

группы #БПМИ224, 2 курса

Черленок Тимофей Игоревич

Принял руководитель проекта:

Виликотский Анатолий Евгеньевич

Менеджер по стратегическим инновациям

"SPLAT Global"

Содержание

Аннотация	4
1 Введение	5
2 Обзор литературы / похожих программных решений	6
3 Первый подход к определению цвета	7
3.1 Изначальная шкала определения цвета	7
3.2 Датасет	8
3.3 Выбор модели	9
3.4 Обучение модели	11
4 Второй подход к определению цвета	12
4.1 Общее описание нового подхода	12
4.2 Детекция зубов	13
4.2.1 Датасет	13
4.2.2 Обучение модели	14
4.3 Сегментация зубов	14
4.3.1 Выбор модели	15
4.4 Оптимизация по времени	17
5 Определение цвета зубов	18
5.1 VITA Bleachguide 3D-MASTER	18
5.2 Выбор цветового пространства	19
5.3 Процесс определения цвета зубов	19
6 Предобработка изображений	20
7 Разработка API приложения	23
7.1 APIController	24
7.2 Документация по использованию API	24
7.3 Другие классы API	25
7.4 Основа базы данных - рекомендательной системы	26
8 Заключение	27

Аннотация

Данная работа будет заключаться в разработке мобильного приложения для операционной системы iOS, позволяющего определять цвет зубов и прикус по фото, которое будет использоваться для заботы о здоровье ротовой полости, получения рекомендаций по уходу и отслеживания динамики изменений.

Ключевые слова

Машинное обучение, медицинская диагностика, фотоанализ, мобильное приложение, прикус и цвет зубов, система рекомендаций

1 Введение

В современном мире забота о здоровье зубов играет ключевую роль в поддержании общего благополучия человека. Однако, несмотря на значительные достижения в стоматологии, доступ к качественной оценке состояния зубов и прикуса остается недоступным для многих. В этом контексте разработка мобильного приложения для определения прикуса и цвета зубов представляет собой важный шаг в направлении повышения доступности и комфортности стоматологических услуг.

Целью данного проекта является создание двух моделей - одной для определения прикуса и другой для определения цвета зубов - а также разработка бэкэнда с системой рекомендаций, интегрированных в мобильное приложение, написанное на языке Swift.

Актуальность данной работы определяется не только повышением доступности оценки состояния зубов, но и улучшением самопонимания пациентов о своем стоматологическом здоровье. Это приложение предоставляет возможность пользователям оценить свое состояние и получить рекомендации относительно дальнейших шагов по уходу за полостью рта.

Данный отчет о проекте написан двумя его участниками - Черленком Тимофеем и Вдовцем Ильей, поскольку два других участника учатся на других курсах других образовательных программ и имеют другие дедлайны.

Нашими результатами работы является функционирующая модель, проводящая сегментацию зубов по фотографии с последующим определением цвета по шкале VITA Bleachguide 3D-MASTER, а так же API мобильного приложения, связанное с моделями и фронтендом, и основой системы рекомендаций. Новизна проекта заключается в комбинации этих элементов в удобном и простом для использования мобильном приложении (которое непосредственно было написано другим участником команды).

Остальные же участники команды занимались фронтендом мобильного приложения и разработкой модели по определению прикуса, но об этом подробнее в следующем разделе

Разделение задач между участниками

Обозначим решаемые задачи и ответственных за них людей:

- Черлёнко Тимофей занимался бэкэндом, разработкой API для мобильного приложения, созданием основы рекомендательной системы, написанием функций по предобработке изображений для определения цвета зубов и выделении цвета на сегментированных картинках, а так же в проведении UX-исследования для определения итогового функционала приложения.

- Вдовец Илья ответственен за написание модели для распознавания полости рта и дальнейшей её сегментации, анализ результатов и оптимизации по скорости и точности, финализацию работы модели на сервере, помощи в деплое модели, выбор шкалы для определения цвета.
- Кислов Максим занимается написанием модели для определения прикуса зубов.
- Вяльцев Павел отвечает за всё, что связано с мобильной разработкой.

2 Обзор литературы / похожих программных решений

- DIANA [7]

Данная программа помогает в анализе количества зубного налёта, выявлении заболеваний ротовой полости и выстраивании коммуникации с пациентом и автоматизации хода приема.

Приложение DIANA не позволяет самостоятельное обследование рта и требует серьезной подготовки от пользователя перед использованием - ему/ей как минимум нужно окрасить зубы индикатором налета перед использованием. В сравнении с этим продуктом наше приложение намного легче в использовании, оно подойдет каждому пользователю, и ему/ей не придется посещать клинику для использования. Также стоит заметить, что эта программа платная.

- Diagnocat [2]

Данная программа представляет собой виртуальный помощник стоматолога в планировании лечения, позволяет следить за процессом лечения зубов, а также выявлять различные заболевания ротовой полости.

Это приложение, как и предыдущее, не позволяет пациенту самому проводить анализы, то есть пользователю всегда нужна помощь врача при использовании, что делает программу более сложной для использования. Также сама программа создана именно для врачей, в то время как наше приложение разработано для самих пациентов, чтобы дать им возможность проводить самодиагностику. Также является платной программой.

- Улыбнись AI [1]

Данная программа проводит стоматологический осмотр в домашних условиях, позволяющий провести диагностику полости рта. Она определяет кариес и деминерализа-

цию эмали - этот функционал отличается от того, который предоставляем мы в нашей программе. В отличие от предыдущих двух вариантов в этом приложении доступен бесплатный тариф.

3 Первый подход к определению цвета

Изначально планировалось разработать модель, которая определяла бы цвет зубов по специальной шкале (про итоговый выбор шкалы написано ниже в секции 5), поэтому изначально решением задачи по определению самого цвета занимался Илья. В этой секции будет описан первый подход и выбор inicialной шкалы с результатами и описанием проблем, из-за которых пришлось по итогу отказаться и от этой inicialной шкалы и от такого подхода по определению цвета.

3.1 Изначальная шкала определения цвета

После изучения вопроса, существует ли специальная шкала для определения цвета зубов в ортодонтии, мы выяснили, что существует шкала **VITA classical A1-D4**, известная всем дантистам. (см. рис 3.1)



Рис. 3.1: Цветовая система VITA classical A1-D4

Шкала VITA Classical A1-D4 является одной из наиболее широко используемых шкал в стоматологии для определения цвета зубов. Она представляет собой систему классификации, разработанную компанией VITA Zahnfabrik в Германии, и используется для стандартизации цвета зубов при изготовлении зубных протезов, коронок, виниров и других стоматологических конструкций.

В шкале VITA Classical A1-D4 используется буквенно-цифровое обозначение для характеристики цвета зубов. Буквенная часть указывает на основной оттенок цвета зуба:

- А обозначает оттенки, близкие к желтому.
- В обозначает оттенки, близкие к желто-красному.
- С обозначает оттенки, близкие к серо-коричневому.
- D обозначает оттенки, близкие к серому.

Цифровая часть указывает на светлоту или темноту цвета зуба внутри данного основного оттенка:

- 1 указывает на самые светлые оттенки.
- 2 на следующие по светлоте.
- 3 на средние по светлоте оттенки.
- 4 на самые темные оттенки.

Таким образом, самый светлый цвет зуба будет А1, а самый темный - D4. Каждый оттенок внутри шкалы имеет свои характеристики, которые определены стандартными образцами.

Выбор на данную шкалу изначально пал, поскольку именно по ней почти все стоматологические компании определяют цвет зубов для выдачи рекомендаций и продажи тех или иных продуктов. Однако основная проблема данной шкалы заключается в её неудобстве использования для простых пользователей, потому что, как можно заметить, она нелинейна и людям, не читая, тяжело понять, что конкретно означает их цвет, потому что, как нам чуть позже объяснили, эта шкала определяет цвет дентина, а не текущий цвет зуба. В конце концов мы перешли к шкале, которая предназначена для определения текущего цвета зуба. (см [рис.5.1](#))

(Работу над этой главой выполнял Вдовец Илья.)

3.2 Датасет

Следующим этапом после определения того, как мы будем определять цвет, встал вопрос о датасете и его дальнейшей разметке, потому что, как и ожидалось, нужного нам размеченного датасета не было. Были пересмотрены пару сайтов с датасетами, где можно было бы хотя бы найти фотографии людей, на которых рот открыт и видны зубы:

- Kaggle [10]
- Roboflow [13]

Что касается Kaggle, то на этом сайте было много подходящих фотографий (суммарно 6998) и они сразу были поделены на выборки (test - 999 фотографий, train - 4999 фотографий и val - 1000 фотографий), но самостоятельно размечать такой датасет полностью с нуля - это непосильная задача, поэтому пришла идея взять и скачать оттуда 17 фотографий и разметить их.

Чтобы разметить датасет, было просмотрено множество специальных программ:

- LabelImg [11]
- Встроенный разметчик датасета на сайте Roboflow [14]
- VoTT [24]

Что касается отдельных программ, таких как LabelImg [11] и VoTT [24], то проблема была в том, что надо было сразу думать о том, какую нейросеть мы будем использовать, потому что многие предобученные модели, например, YOLOv8 [18] или любая из моделей из TensorFlow 2 Detection Model Zoo [28] требует особый формат размеченных данных, например у YOLOv8 [18] он так и называется и эта модель принимает только этот формат данных, а если говорить про модели из TensorFlow 2 Detection Model Zoo [28], то там подойдут форматы COCO [3] и TFRecord [23]. Однако на сайте Roboflow есть разметчик и встроенные скрипты, которые позволяют экспортировать размеченный датасет в любом известном формате (см. рис 3.2)

В конце концов, изначальный датасет был размечен на сайте Roboflow [6]

Однако по ходу разметки из-за неверного выбора изначальной шкалы, было трудно на глаз определять цвет, из-за чего возникли проблемы с разметкой. Поэтому для простоты понимания были размечены только цвета A1-A4.

(Работу над этой главой выполнял Вдовец Илья.)

3.3 Выбор модели

К выбору модели мы подошли со всей ответственностью, поскольку проект нужно было сдать в довольно жесткие сроки и этот проект делался в промышленных целях, то времени на то, чтобы писать свою архитектуру, не было. Идеальным решением было использовать предобученную модель и дообучить ей на своём датасете. Было просмотрено

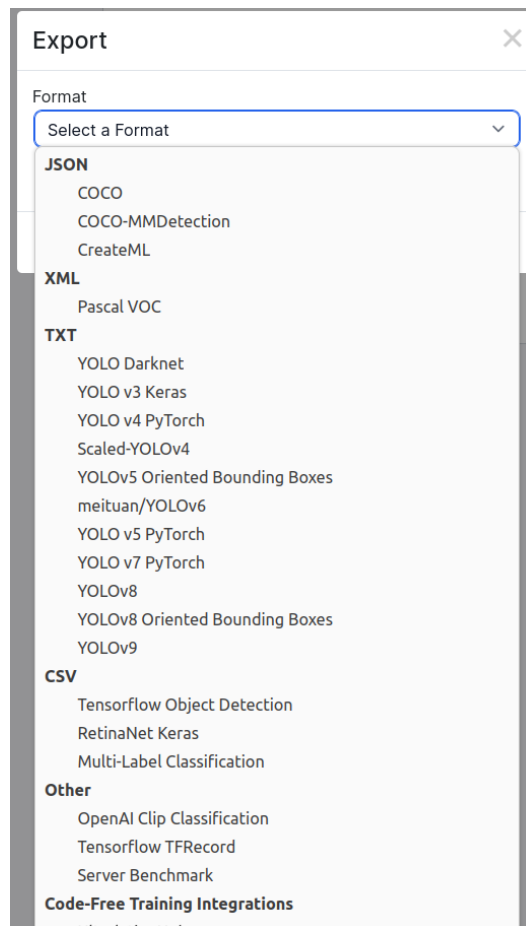


Рис. 3.2: Возможность экспортировать датасет с Roboflow в любом формате

множество моделей, решающих задачу object detection. (см выше). В итоге для MVP было принято решение использовать YOLOv8 [19].

Немного про YOLOv8

YOLO (You Only Look Once), популярная модель обнаружения объектов и сегментации изображений, была разработана Джозефом Редмоном и Али Фархади в Университете Вашингтона. Появившись в 2015 году, YOLO быстро завоевала популярность благодаря своей высокой скорости и точности.

YOLOv8 Это последняя версия YOLO от Ultralytics. Являясь передовой, современной (SOTA) моделью, YOLOv8 опирается на успех предыдущих версий, представляя новые возможности и улучшения для повышения производительности, гибкости и эффективности. YOLOv8 поддерживает полный спектр задач ИИ зрения, включая обнаружение, сегментацию, оценку позы, отслеживание и классификацию. Такая универсальность позволяет пользователям использовать возможности YOLOv8 в различных приложениях и областях.

(Работу над этой главой выполнял Вдовец Илья.)

3.4 Обучение модели

Ещё одним плюсом Roboflow является возможность скачивания датасета с помощью Roboflow SDK для Python.

```
1 # pre-requirements - pip install roboflow
2 from roboflow import Roboflow
3 def download_dataset():
4     rf = Roboflow(api_key="oQfzohj8A9yzztxMDSFd")
5     project = rf.workspace("vita-dental-color-dataset").project("vita-colour-detecti
6     version = project.version(2)
7     dataset = version.download("yolov8")
```

Для дальнейшего обучения, обратимся к официальной документации Ultralytics [19], которые предоставляют модель семейства YOLO. Обучение происходило в Google Colab, потому что встроенной хорошей видеокарты у меня не было. Главным неудобством было то, что доступ к их GPU (см. рис 3.3 давался только на сутки, а потом надо было ждать, пока пройдет ещё день, чтобы можно было снова обучать, поэтому обучать надо было оперативно.

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
NVIDIA-SMI 510.47.03			Driver Version: 510.47.03				CUDA Version: 11.6		
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
GPU	Name	Persistence-M		Bus-Id	Disp.A	Volatile	Uncorr. ECC		
Fan	Temp	Perf	Pwr:Usage/Cap		Memory-Usage	GPU-Util	Compute M.	MIG M.	
=====									
0	Tesla T4		Off	00000000:00:04.0	Off			0	
N/A	64C	P0	31W / 70W		0MiB / 15360MiB	0%	Default	N/A	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
Processes:									
GPU	GI	CI	PID	Type	Process name	GPU Memory			
	ID	ID				Usage			
=====									
No running processes found									
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									

Рис. 3.3: GPU на сервере Google

У Ultralytics есть очень удобный YOLO CLI, который очень облегчает процесс обучения.

Как видно по результатам, модель очень сильно путается в цветах и можно было подумать, что дело в размере датасета, но повторюсь, что при разметке проблема была в том, чтобы самому понять, того ли цвет зуба или другого.

Если посмотреть на результат предсказания, то получим то, что и ожидали:

(Работу над этой главой выполнял Вдовец Илья.)

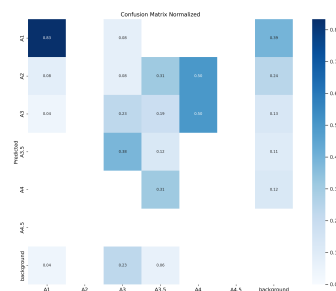


Рис. 3.4: Confusion matrix

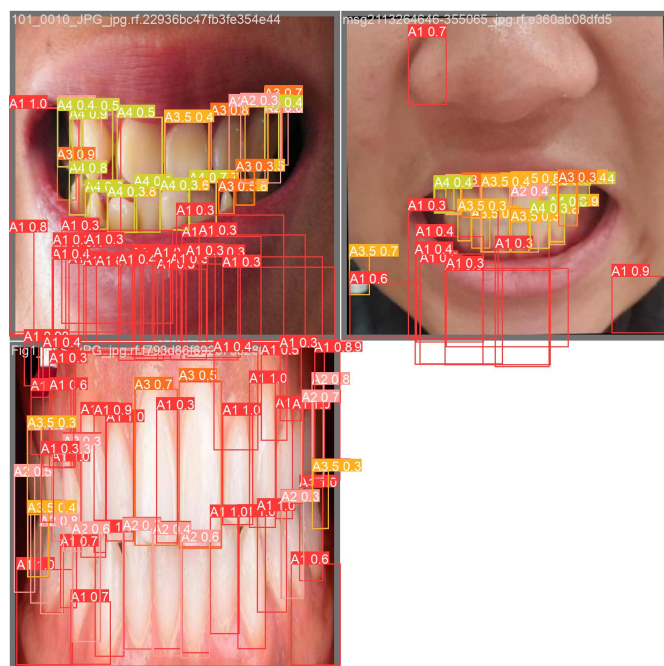


Рис. 3.5: Результаты предсказаний исходной модели

4 Второй подход к определению цвета

Теперь перейдём ко второму подходу, который пришел на смену старому и решил все его проблемы.

4.1 Общее описание нового подхода

Суммируя все проблемы предыдущего подхода, можно выделить 2 явные проблемы:

- 1 Нелинейная шкала
- 2 Ручная разметка цветов ведёт к неточному определению цвета

Первую проблему было решить довольно просто, поскольку надо было просто поменять шкалу на линейную, благо компания VITA имеет таковые и более подробно про

выбранную шкалу будет написано чуть позже. (см. раздел 5.1) А что касается второй проблемы, то тут уже встал вопрос о том, стоит ли делать задачу именно определения цвета статистической. Что имеется в виду под этим?

Раньше мы пытались определить цвет с помощью модели, а модель - это вещь, использующая внутри себя множество статистических алгоритмов, которые используют такие понятия как вероятность и точность, что и даёт неопределённость в результатах. Хотелось бы в идеале использовать детерминированные алгоритмы для определения цвета, то есть получать цвет зуба уже на основе не какой-то вероятности, а какой цвет зуба есть, такой и найти. Другими словами, чтобы ошибка была равна $\mathcal{O}(\epsilon_{machine})$

Наша команда придумала, как это сделать. Поделим теперь задачу определения цвета зубов на 3 части.

- 1 Детекция зубов в ортодонтическом формате названий (чтобы можно было использовать эти результаты в дальнейшем)
- 2 Сегментация найденных зубов
- 3 Определение цвета каждого из зубов

Про определение цвета можете прочитать чуть ниже в разделе 5. Что касается решения первых двух задач, то видно, что нам понадобится две модели - по детекции и сегментации соответственно.

Для начала решим задачу teeth detection.

(Работу над этой главой выполнял Вдовец Илья.)

4.2 Детекция зубов

Поскольку в первом подходе по сути решалась задача color detection, то решено было использовать ту же предобученную модель, но обучить её на другом датасете - речь идёт про YOLOv8. [19]

4.2.1 Датасет

В этот раз найти датасет было гораздо проще, потому что после первого подхода было понятно, где искать и как размечать в случае, если это придётся. Однако, нам повезло, и на тему задачи teeth detection было довольно много уже готовых датасетов с разным типом разметки, начиная от типов кариеса, заканчивая различными видами межзубных образований.

Нам повезло и мы обнаружили идеально подходящий датасет с нужной разметкой [5]

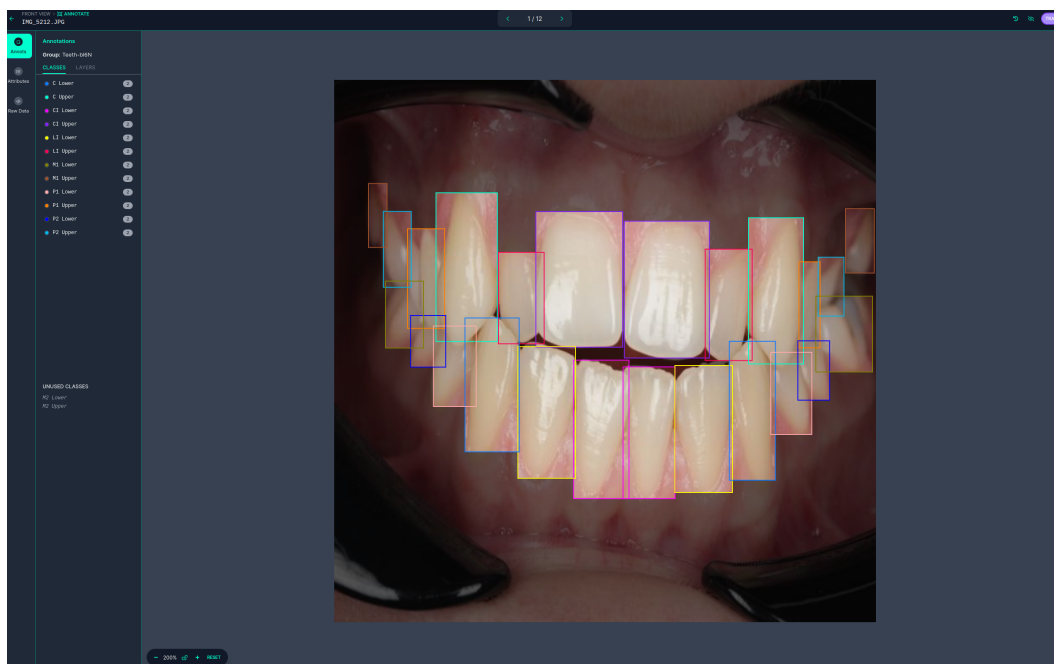


Рис. 4.1: Пример разметки в новом датасете

Тут и фотографий уже побольше - 58 на train, 18 на validation и 8 на test

Этот датасет будет определять зубы вот в таком формате [27], что безумно удобно в ортодонтии.

(Работу над этой главой выполнял Вдовец Илья.)

4.2.2 Обучение модели

Полностью аналогично разделу 3.4, только обучение будет происходить на другом датасете.

Давайте просто сразу посмотрим на результаты обучения:

Сразу видно, что дело обстоит намного лучше, чем было до этого. Видно, что отклонения есть, но они не столь существенны и отражают нормальные результаты статистической работы модели.

Можно проверить и убедиться в совершенно ином качестве предсказаний, нежели это было в первом подходе:

(Работу над этой главой выполнял Вдовец Илья.)

4.3 Сегментация зубов

Теперь, когда мы знаем bounding box, в которых находятся наши зубы, хотелось бы знать их точную форму и уметь возвращать каждый зуб в формате фотографии.

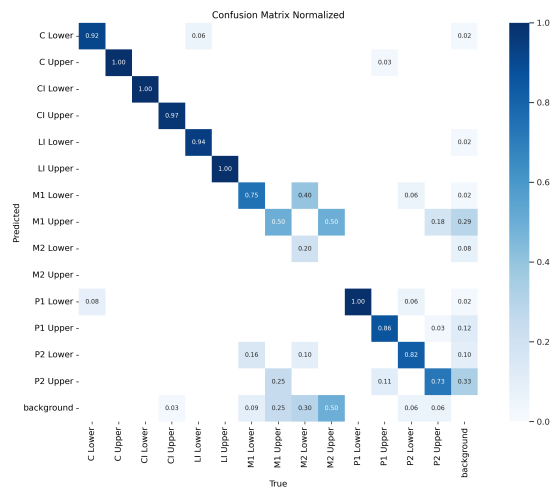


Рис. 4.2: Confusion matrix



Рис. 4.3: Примеры предсказаний при детекции зубов

4.3.1 Выбор модели

На самом деле задача сегментации являлась основной и самой главной, потому что потом каждая фотография зуба будет использована для того, чтобы определить цвет(более подробно про сам алгоритм, можно прочитать в разделе 5), поэтому надо было выбрать модель так, чтобы если что пожертвовать немного временем, потому что потом все равно пришлось бы оптимизировать, но преуспеть в точности сегментации. Нами был проведён ресерч и найдены несколько вариантов подходящих моделей:

- Yolov9 [20]
- SAM [17] [22]

- FastSAM [29] [15]
- MobileSAM [16]

Стоит понимать, что при обучении на датасете самому ты всегда получишь результат хуже, чем это сделают за тебя в больших компаниях, на невероятно мощных серверах, обучая на миллиардах фотографий, поэтому было решено не использовать модель YOLOv9 [20]. Нам удалось найти модель, способную сегментировать что угодно без самостоятельного обучения.

SAM

SAM — это модель от Meta AI, обученная на новом огромном датасете для сегментации изображений. Этот датасет содержит 11 млн картинок и 1.1 млрд масок сегментации. Meta AI описали, как собирали этот датасет, а также выложили его в открытый доступ вместе с моделью.

SAM — очень мощная модель. SAM способна на одном изображении сегментировать сразу множество объектов, включая объекты небольшого размера и даже те, что модель не видела во время обучения.

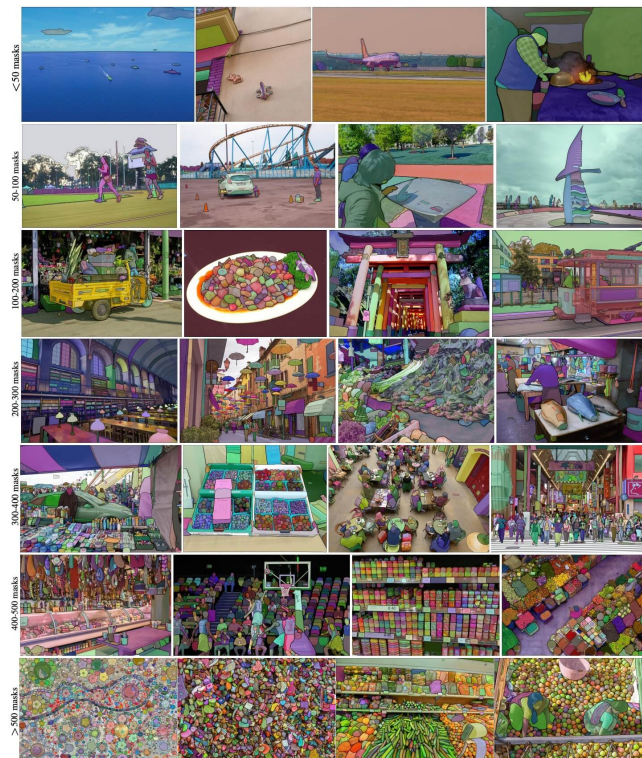


Рис. 4.4: Пример работы SAM

Если интересно прочитать подробнее, то оставляю ссылку на статью [22]

Опять-таки Ultralytics предоставляет наудобнейшие методы и доступ к этой модели, поэтому использовать её было одним удовольствием.

Однако конкретно у SAM был минус во времени, потому что по точности она не сильно уступает FastSAM, но FastSAM в 50 раз быстрее. Так что использовать было принято решение его. Что касается MobileSAM, то эта модель уже сильно менее точна, и как я сказал, нам точность важнее, чем время.

Теперь, когда нам поступает на вход bounding box, в которой находится зуб, то подрубается FastSAM и находит внутри box всё, что есть и сегментирует, но хитрость заключается в том, что внутри bounding box нет ничего, кроме зуба, поэтому вся точность модели уходит только на сегментацию зуба, а это в точности то, что нам и было нужно. Далее, просто после вернувшейся бинарной маски зуба, мы применяем эту маску к фотографии исходной и вырезаем наш зуб и кладём в массив к остальным зубам. Таким образом, мы получаем массив из фоток каждого зуба и остается только определить цвет по новой шкале. (об этом чуть ниже)

(Работу над этой главой выполнял Вдовец Илья.)

4.4 Оптимизация по времени

У нас теперь есть модель, которая находит и сегментирует зубы на фотографии и возвращает массив из зубов, однако изначальное время на обработку одной фотографии очень сильно зависело от размера самой фотографии, что было просто ужасно и это срочно надо было нормализовать. Нашей командой было перепробовано несколько размеров исходных фотографий и лучше всего модель показала себя на размере исходной фотографии 320×320 , поэтому написали простую функцию в бэкэнде для сжатия изображения:

```
1 def resize_image(images):
2     resized_images = []
3     for image in images:
4         resized_image = cv.resize(image, (IMAGE_SIZE, IMAGE_SIZE))
5         resized_images.append(resized_image)
6     return resized_images
```

Например, до сжатия изображения, время на обработку фотографии 1024×1024 была примерно минута, а после сжатия стала 11 секунд.

Ещё одной оптимизацией стало количество ядер, потому что при количестве ядер, меньшее, чем 5, FastSAM-х начинала вести себя медленно (около минуты на запрос), поэтому было принято решение пожертвовать точностью и использовать FastSAM-s в случае меньшего количества ядер. (эта версия FastSAM более легковесная и оптимизирована для работы на более слабых ПК). На сервере, на котором работает, наше приложение как раз

стоит 2 ядра, поэтому данное оптимизационное решение было существенным, так как оно позволило вернуть время на обработку одной фотографии на 6-11 секунд.

(Работу над этой главой выполнял Вдовец Илья.)

5 Определение цвета зубов

Следующей задачей разработки после сегментации зубов стояло непосредственно определение цвета зубов. После того, как мы отказались от использования модели для определения цвета, эта задача перешла в бэкенд часть проекта.

5.1 VITA Bleachguide 3D-MASTER

После нескольких кругов обсуждений с врачами компании ООО "СПЛАТ ГЛОБАЛ" того, какая цветовая система лучше подойдет для нашего проекта, и написания пробных версий проекта мы вместе пришли к решению остановиться на цветовой системе VITA Bleachguide 3D-MASTER (рис 5.1), причины чего описаны выше. Суть системы заключается в том, что у цвета зуба выделяются три характеристики: светлота цвета, интенсивность цвета, и тон цвета.

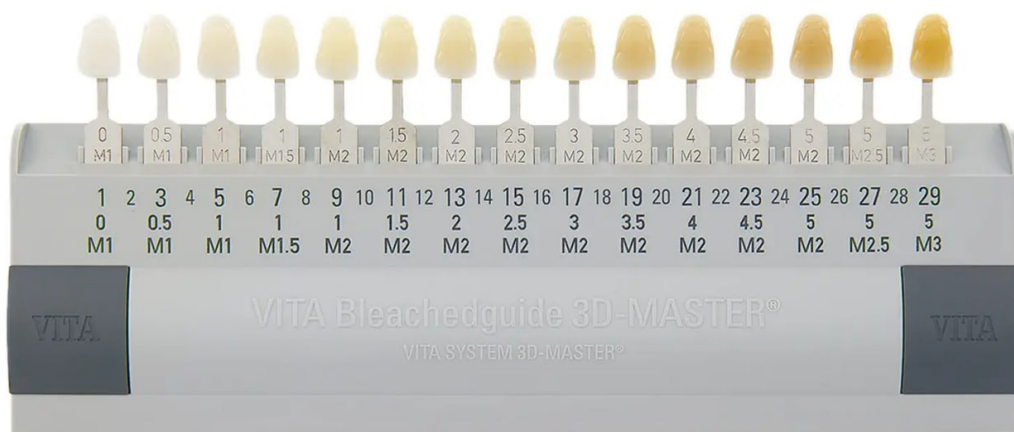


Рис. 5.1: Цветовая система VITA Bleachguide 3D-MASTER

Таким образом, теперь перед нами стояла задача для сегментированных фотографий зубов провести классификацию по системе VITA Bleachguide 3D-MASTER. Для этого нужно было для каждого зуба определить, к какому цвету из системы он ближе расположен в спектре.

5.2 Выбор цветового пространства

По умолчанию библиотека OpenCV [21], которой мы пользовались для работы с изображениями, принимает, хранит передает и обрабатывает изображения в пространстве BGR (тот же RGB, только другой порядок цветов) и мы решили изучить, является ли использование этого цветового пространства лучшим вариантом для классификации цвета на изображениях сегментированных зубов.

Обратившись к научным статьям по этой теме, мы наткнулись на научную работу 'Experimental comparison of color spaces for material classification' [4], опубликованную в журнале Journal of Electronic Imaging, в которой проводится сравнительный анализ различных цветовых пространств относительно их использования для задач классификации. Авторы статьи провели сравнительный анализ 10 различных цветовых пространств, и результатом проделанной работы стало заключение, что пространство CIELAB значительно лучше других цветовых пространств для задач классификации. Ознакомившись с результатами анализа, проведенного в данной научной статье, мы пришли к решению использовать именно это цветовое пространство - CIELAB.

CIELAB (другое название - CIE 1976 $L^*a^*b^*$) - цветовое пространство, разработанное специально с учетом человеческого восприятия цвета в попытке сделать изменение цвета более линейным и его восприятие не будет зависеть от области цветового пространства, в котором оно происходит. Главными составляющими цвета в этом пространстве являются три координаты: L - светлота, которая измеряется от 0 до 100, а - положение цвета в диапазоне от зелено-голубого до красно-малинового и b - положение цвета в диапазоне от голубого до желтого. И a, и b лежат в диапазоне от -128 до 127.

5.3 Процесс определения цвета зубов

Теперь перейдем к рассмотрению реализации самого алгоритма определения цвета зубов.

От обученной модели машинного обучения мы получаем два массива с сегментированными картинками зубов с двух фотографий, предоставленных пользователем. Далее для каждого зуба мы находим среднее значение его цвета, переведенное из пространства BGR в CIELAB. Мы сделали преподсчет цветовых значений CIELAB для каждого из эталонных зубов цветовой системы, благодаря чему для каждого зуба пользователя мы можем быстро

находить Евклидову метрику с каждым эталонным зубом по формуле

$$\Delta E = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}, \quad (1)$$

где (L_1, a_1, b_1) – параметры зуба, для которого хотим определить цвет, а (L_2, a_2, b_2) – параметры любого эталонного зуба.

Таким образом, цвет зуба среди эталонных, который находится на наименьшем Евклидовом расстоянии от изучаемого, и является предполагаемым цветом зуба пациента.

Следующим шагом работы алгоритма является сравнение результата, полученного для одних и тех же зубов с двух разных фотографий. Если результаты не совпадают, то мы считаем данный зуб выбросом и не учитываем его. Среди зубов, которые проходят это отсеивание мы выбираем тот цвет, который встречается чаще всего и выдаем его пользователю.

Таким образом, мы получили готовый алгоритм определения цвета зубов. Для тестирования работы алгоритма мы провели сравнение результата, который дало наше приложение, с заключением эксперта по цвету зубов на выборке из качественно сделанных снимков. Результат в 86% правильных результатов выданных моделью мы считаем удачным.

Также стоит заметить, что наш алгоритм решает более сложную задачу, чем та, что была поставлена нам компанией ООО "СПЛАТ ГЛОБАЛ": он определяет не просто цвет зубов, а обозначает цвет каждого зуба. На данный момент такая детализация исследования не нужна компании ООО 'СПЛАТ ГЛОБАЛ', но в будущих версиях приложения она уже может быть внедрена в функционал.

(Работу над этой главой выполнял Черленок Тимофей.)

6 Предобработка изображений

После создания первого прототипа приложения и первого рабочего варианта алгоритма по определению цвета, мы решили провести UX-исследование среди наших потенциальных пользователей, чтобы протестировать удобность интерфейса и, возможно, найти какие-то вызывающие проблемы у пользователей элементы приложения. Для этого мы отобрали среди своих знакомых людей, которые проявили заинтересованность в использовании приложения для самодиагностики состояния ротовой полости (именно такой критерий определения нашей целевой аудитории мы поставили). Во время тестирования мы не выявили никаких проблем непосредственно с использованием приложения, но внезапно для самих себя заметили интересную закономерность, о которой не задумывались раньше.

Получив доступ к изображениям, которые отправляли на обработку наши потенциальные пользователи, мы заметили, что люди часто отправляют довольно темные фотографии. Мы решили проверить, влияет ли это на итоговый результат классификации цвета, который выдает наше приложение.



Рис. 6.1: Оригинал фотографии зубов, для которой производилось затемнение

Мы собрали ряд изображений, для которых нам известна экспертная оценка цвета зубов и искусственно их затемнили для создания эффекта, будто фотография была сделана в темном помещении при плохом освещении (примеры представлены на рисунках [6.1](#) и [6.2](#))

После этого мы провели анализ каждой фотографии, используя наше приложение, и выявили, что в 6 из 10 случаев результат анализа для затемненной фотографии отличался от результата для оригинала фото, который во всех 10 случаях совпадал с ожидаемым. Например, для фотографии из первого рисунка с оригиналом фотографии зубов приложение выдавало результат М3-5 по шкале VITA SYSTEM 3D-MASTER, что является правильным цветом зубов на картинке, в то время как результатом для второго, затемненного изображения стал цвет М2-5 по той же шкале. Данный результат является подтверждением нашей гипотезы о влиянии яркости изображения на результат анализа цвета зубов.

Тогда мы решили разработать алгоритм предобработки фотографий зубов, который бы позволил нам избегать неточности анализа, вызванного плохим освещением на фотографии.

Первым шагом для этого стало собственно определение того, какое освещение на фотографии считается нормальным. Для этого мы собрали выборку из 15 фотографий, обре-



Рис. 6.2: Затемненная нами фотография зубов

занных так, что на фотографии присутствует только рот человека, ведь нам важна освещенность именно этой части фотографии для анализа, каждая из которых визуально выглядела достаточно освещенной, что специалист может по ней определить цвет зубов, и при этом каждая из этих фотографий получала правильное значение цвета зубов при анализе через наше приложение. Посредством изменения режима изображений на grayscale, используя функционал библиотеки OpenCV [21], и подсчета среднего значения цвета по пикселям мы смогли найти среднее значение "яркости которую должно иметь наше изображение для точного определения цвета зубов. Что неудивительно, среднее значение составило примерно 128.6, что является практически серединой диапазона глубины цветовых компонентов, который содержит в себе значения от 0 до 255. Это означает, что для корректного определения цвета зубов яркость изображения должна быть средней - изображение хорошо анализируется, если оно не слишком темное, но и не слишком светлое.

Далее мы написали функцию, применяющую фильтрацию, нормализующую яркость для входных изображений.

Алгоритм работы этой функции заключается в создании копии исходного изображения в режиме grayscale и нахождении его среднего значения "яркости используя которое мы можем найти коэффициент, на который нужно отнормировать исходное изображение, после чего собственно и выполняем нормировку.

На рисунке 6.3 представлен результат предобработки затемненного изображения с рисунка 6.2, как мы видим изображение стало практически неотличимым от оригинала.



Рис. 6.3: Затемненная нами фотография зубов после прохождения через функцию по нормализации яркости

Это также подтверждается и при использовании улучшенного изображения для определения цвета зубов. В этот раз программа стала выдавать правильный результат - цвет М3-5.

Для тестирования работы новой функции мы провели такую же работу с остальными изображениями из этой выборки и получили, что программа стала выдавать правильный результат во всех случаях, а также использовали фотографии, которые получили от пользователей во время юзабилити-тестирования и получили, что программа работает и правильно определяет цвет зубов в 81% случаев, что мы считаем довольно хорошим результатом.

Таким образом, функция предобработки фотографий позволила нам сильно улучшить точность работы программы.

(Работу над этой главой выполнял Черленок Тимофей.)

7 Разработка API приложения

В данном разделе подробно описана структура бэкенда приложения.

Для написания API приложения было принято решение использовать язык Python [8] и веб-фреймворк Flask [26] из-за их удобства и скорости работы.

7.1 APIController

Главным классом API является класс `APIController`, содержащий в себе три главных метода:

- `handle_photo_upload` - метод, проверяющий, загрузил ли пользователь нужное для данного типа анализа (для анализа прикуса требуются 3 фотографии, а для анализа цвета требуются всего две фотографии)
- `handle_analysis_request` - метод, совершающий анализ по фотографиям. Для этого производится декодирование фотографий из JSON, полученного в URL и преобразование их в NumPy [25] массивы. После совершается вызов соответствующего метода у поля класса - объекта класса `PhotoAnalysisService`, о котором будет написано ниже, и передача в него декодированных фотографий, который и возвращает результат анализа изображений.
- `provide_recommendations` - метод, выдающий рекомендации по уходу за полостью рта на основе результатов анализа. Он получает результат анализа из JSON, полученного в URL и создает объект класса `RecommendationService`, о котором будет написано ниже, после чего вызывает функцию этого класса для получения рекомендаций и возвращает их.

Каждая из этих функций после используется вместе с декораторами, которые задают нашу веб-фреймворку, какие URL нужно использовать для работы с этими методами.

7.2 Документация по использованию API

Мы используем сервер компании ООО 'СПЛАТ ГЛОБАЛ' , на котором мы разместили API приложения. Вот список функций для работы с API:

- `http://адрес сервера/handle_photo_upload_color` - на данный URL адрес можно отправить POST-запрос, передав в него JSON с двумя полями - `user_photo_0` и `user_photo_1`, каждое из которых содержит строку, состоящую из байтов изображения, закодированных с помощью base64. (Полный пример запроса не будем приводить в отчете из-за слишком большой длины кодированных изображений-строк) Произойдет проверка фотографий на соответствие требованиям для анализа цвета. При корректной работе вернется изначальный JSON

- `http://адрес сервера/handle_photo_upload_bite` - аналогичная функция, выполняющая проверку на соответствие требованиям для анализа прикуса. JSON в этом случае содержит три поля - `user_photo_0`, `user_photo_1`, `user_photo_2`
- `http://адрес сервера/handle_analysis_request_color` - на данный URL адрес можно отправить POST-запрос, передав в него JSON с двумя полями - `user_photo_0` и `user_photo_1`, каждое из которых содержит строку, состоящую из байтов изображения, закодированных с помощью `base64`. Метод проведет анализ цвета и текстом ответа будет являться результат анализа.
- `http://адрес сервера/handle_analysis_request_bite` - на данный URL адрес можно отправить POST-запрос, передав в него JSON с тремя полями - `user_photo_0`, `user_photo_1`, `user_photo_2`, каждое из которых содержит строку, состоящую из байтов изображения, закодированных с помощью `base64`. Метод проведет анализ прикуса и текстом ответа будет являться результат анализа.
- `http://адрес сервера/provide_recommendations_color` - на данный URL адрес можно отправить POST-запрос, передав в него JSON с полем - `analysis_result`, содержащим строку с результатом анализа. Метод вернет JSON, содержащий рекомендации по уходу и результат цвета.
- `http://адрес сервера/provide_recommendations_bite` - на данный URL адрес можно отправить POST-запрос, передав в него JSON с полем - `analysis_result`, содержащим строку с результатом анализа. Метод вернет JSON, содержащий рекомендации по уходу и результат прикуса.

7.3 Другие классы API

Следующим классом API является класс `PhotoAnalysisService`, который включает в себя два метода - `analyze_color` и `analyze_bite`, внутри которых происходит создание объекта соответствующего класса модели и вызов метода анализа фотографии.

Класс `BiteAnalysisModel` написан как каркас для последующей реализации Максимом Кисловым, класс включает в себя метод `analyze_photo`, внутри которого Максиму нужно будет прописать обращение к своей модели по анализу прикуса.

Класс `ColorAnalysisModel` уже закончен и имеет более целостную структуру. Он включает в себя поле `teeth_classification` - список CIELAB параметров для каждого из эталонных зубов шкалы VITA Bleachguide 3D-MASTER, предподсчитанных нами заранее, а так же

несколько методов: `improve_brightness`, работа которого описана в главе 6, `analyze_photo` и `get_color`, работа которых описана в главе 5, а также функция `resize_image`, внутри которой происходит изменение размеров изображения.

Последним классом является класс `RecommendationService`, который при создании нового объекта выполняет подключение к базе данных, используя функционал библиотеки `sqlite3` [9], а еще включает в себя два метода:

- `fetch_recommendations_color` - метод, выполняющий SQL-запрос `'SELECT description, products from colour_recommendations WHERE name == ?'` к базе данных для получения рекомендаций по уходу за полостью рта в зависимости от результата анализа цвета.
- `fetch_recommendations_bite` - метод, выполняющий SQL-запрос `'SELECT description, products from recommendations WHERE id == ?'` к базе данных для получения рекомендаций по уходу за полостью рта в зависимости от результата анализа прикуса.

7.4 Основа базы данных - рекомендательной системы

Для создания основы рекомендательной системы (после обсуждения этого вопроса со специалистами компании ООО "СПЛАТ ГЛОБАЛ" мы пришли к решению, что мы предоставим компании основу базы данных, которую врачи компании заполняют рекомендациями сами) мы создали базу данных с двумя таблицами внутри нее - `recommendations` и `colour_recommendations` (см рисунки 7.1 и 7.2).

id	name	description	products
1	Дистальный прикус	Дистальный прикус	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
2	Мезиальный прикус	Мезиальный прикус	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
3	Глубокий прикус	Глубокий прикус	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
4	Открытый прикус	Открытый прикус	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
5	Перекрестный прикус	Перекрестный прикус	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
6	Нормальный прикус	Нормальный прикус	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'

Рис. 7.1: Основа базы данных - Таблица `recommendations` для прикуса

Таблица `colour_recommendations` содержит следующие атрибуты: `id` - идентификатор цвета, первичный ключ таблицы, `description` - небольшое описание-расшифровка значения цвета зубов, `name` - значение результата анализа цвета, и `products` - список рекомендованных продуктов в виде JSON. Врачам из компании ООО "СПЛАТ ГЛОБАЛ" требуется самостоятельно ввести значения `description` и `products`.

Таблица `recommendations` содержит такие же атрибуты.

id	description	name	products
1	Ваши зубы очень белые	M1-0	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
2	Ваши зубы очень белые	M1-0.5	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
3	Ваши зубы очень белые	M1-1	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
4	Ваши зубы белые	M1.5-1	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
5	Ваши зубы белые	M2-1	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
6	Ваши зубы белые	M2-1.5	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
7	Ваши зубы немного жел...	M2-2	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
8	Ваши зубы немного жел...	M2-2.5	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
9	Ваши зубы немного жел...	M2-3	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
10	Ваши зубы желтые	M2-3.5	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
11	Ваши зубы желтые	M2-4	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
12	Ваши зубы желтые	M2-4.5	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
13	Ваши зубы очень желтые	M2-5	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
14	Ваши зубы очень желтые	M2.5-5	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'
15	Ваши зубы очень желтые	M3-5	'["Зубная паста", "Зубная нить", "Ополаскиватель для рта"]'

Рис. 7.2: Основа базы данных - Таблица *colour_recommendations*

Сама база данных хранится в файле db формата на сервере.

Для обращения к базе данных используется язык программирования SQL [12].

(Работу над этой главой выполнял Черленок Тимофей.)

8 Заключение

В этом разделе будут описаны результаты нашей работы. Мы не будем подробно описывать результаты, полученные другими участниками нашей команды, а именно, результаты по созданию фронтенда мобильного приложения и модели по определению прикуса.

В результате проекта была разработана модель по сегментации ротовой полости по фотографиям пользователей, которая способна сегментировать зубы и возвращать массив изображений зубов, мы получили важный инструмент для автоматизации анализа стоматологических изображений. Эта модель обладает значительным потенциалом для применения в различных областях стоматологии, таких как диагностика, планирование лечения и мониторинг заболеваний полости рта.

Основные достоинства предложенной модели включают в себя точность сегментации, скорость работы и возможность выдавать результат в специальной стоматологической шкале, что поможет врачам легче сориентировать пользователей по лечению и рекомендациям. Ее способность с высокой точностью выделять зубы на фотографиях пользователей является ключевым преимуществом, обеспечивающим высокую степень надежности в процессе анализа.

Однако, несмотря на эти преимущества, модель также имеет некоторые ограничения. Например, ее эффективность может быть снижена при работе с изображениями низкого качества или в условиях непредвиденных артефактов. Кроме того, для более широкого применения модели может потребоваться дальнейшая оптимизация и адаптация к различным клиническим сценариям.

Еще один главный результат работы - создание алгоритма определения цвета зубов вместе с алгоритмом предобработки изображений для анализа. Нам удалось реализовать алгоритм, который может быстро проводить анализ состояния полости рта пациента без вмешательства профессионала. Именно возможность проведения исследования без зубного врача выделяет нашу работу среди похожих приложений и является большим плюсом для обычных пользователей. Дальнейшим развитием проекта может являться работа над улучшением точности выдаваемого результата путем применения более продвинутых алгоритмов.

Другим важнейшим результатом проекта стало создание API для мобильного приложения, которое позволило связать фронтенд приложения с реализованными моделями машинного обучения. API удобно в использовании, позволяет быстро обрабатывать запросы. Создание API является важным этапом для достижения цели сделать процесс диагностики полости рта доступным для пользователей без присутствия врача. Перспективой дальнейшей работы может являться создание системы защиты сервера, более продвинутой системы обработки исключений, которые могут понадобиться при сильном масштабировании проекта.

И еще одним важным результатом нашей работы стало создание основы рекомендательной системы в виде базы данных, которую врачи компании в дальнейшем смогут самостоятельно наполнить качественными советами по уходу за полостью рта и рекомендациями товаров, а так же выполнять обновления системы на постоянной основе.

В заключении прилагаем ссылку на репозиторий с нашим [исходным кодом](#).

Список литературы

- [1] ООО 'УЛЫБНИСЬАИ'. *Улыбнись AI*. URL: <https://navigator.sk.ru/orn/1125648> (дата обр. 26.04.2024).
- [2] ООО «ДИАГНОКАТ». *Diagnocat*. URL: <https://diagnocat.ru/> (дата обр. 26.04.2024).
- [3] Amazon. *COCO*. URL: <https://docs.aws.amazon.com/rekognition/latest/customlabels-dg/md-coco-overview.html> (дата обр. 26.04.2024).
- [4] Raquel Bello-Cerezo, Francesco Bianconi, Antonio Fernández, Elena González и Francesco Di Maria. "Experimental comparison of color spaces for material classification". В: *Journal of Electronic Imaging* 25.6 (2016).
- [5] University of Bristol. *Roboflow Dataset*. URL: <https://universe.roboflow.com/university-of-bristol-htbdh/front-view-g47wa/dataset/3> (дата обр. 26.04.2024).
- [6] Vita Dental Color Dataset. *Vita colour detection Dataset*. URL: <https://universe.roboflow.com/vita-dental-color-dataset/vita-colour-detection> (дата обр. 26.04.2024).
- [7] DIANA. *DIANA*. URL: <https://dianapp.online/> (дата обр. 26.04.2024).
- [8] Python Software Foundation. *Python*. URL: <https://docs.python.org/3/> (дата обр. 26.04.2024).
- [9] Python Software Foundation. *sqlite3*. URL: <https://docs.python.org/3/library/sqlite3.html> (дата обр. 26.04.2024).
- [10] Vũ Tun Hi. *Braces and Teeth*. URL: <https://www.kaggle.com/datasets/vutuanhai237/braces-and-teeth> (дата обр. 26.04.2024).
- [11] HumanSignal. *LabelImg*. URL: <https://github.com/HumanSignal/labelImg> (дата обр. 26.04.2024).
- [12] IBM. *SQL*. URL: <https://www.ibm.com/docs/en/db2/11.5?topic=fundamentals-sql> (дата обр. 26.04.2024).
- [13] Roboflow Inc. *Roboflow*. URL: <https://docs.roboflow.com/> (дата обр. 26.04.2024).
- [14] Roboflow Inc. *Roboflow Universe*. URL: <https://universe.roboflow.com/> (дата обр. 26.04.2024).
- [15] Ultralytics Inc. *FastSAM*. URL: <https://docs.ultralytics.com/models/fast-sam/> (дата обр. 26.04.2024).

- [16] Ultralytics Inc. *MobileSAM*. URL: <https://docs.ultralytics.com/models/mobile-sam/> (дата обр. 26.04.2024).
- [17] Ultralytics Inc. *SAM*. URL: <https://docs.ultralytics.com/models/sam/> (дата обр. 26.04.2024).
- [18] Ultralytics Inc. *YOLOv8*. URL: <https://docs.ultralytics.com/models/yolov8/> (дата обр. 26.04.2024).
- [19] Ultralytics Inc. *YOLOv8 Docs*. URL: <https://docs.ultralytics.com/ru> (дата обр. 26.04.2024).
- [20] Ultralytics Inc. *YOLOv9*. URL: <https://docs.ultralytics.com/models/yolov9/> (дата обр. 26.04.2024).
- [21] Intel. *OpenCV*. URL: <https://docs.opencv.org/4.x/index.html> (дата обр. 26.04.2024).
- [22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár и Ross Girshick. *Segment Anything*. Под ред. arXiv.org. 2001. Апр. 2023. URL: <https://arxiv.org/pdf/2304.02643> (дата обр. 26.04.2024).
- [23] Microsoft. *TFRecords*. URL: <https://learn.microsoft.com/ru-ru/azure/databricks/machine-learning/load-data/ddl-data> (дата обр. 26.04.2024).
- [24] Microsoft. *VoTT*. URL: <https://github.com/microsoft/VoTT> (дата обр. 26.04.2024).
- [25] Travis Oliphant. *NumPy*. URL: <https://numpy.org/doc/> (дата обр. 26.04.2024).
- [26] Armin Ronacher. *Flask*. URL: <https://flask.palletsprojects.com/en/3.0.x/> (дата обр. 26.04.2024).
- [27] Carter Carter Solicitors. *Tooth Map*. URL: <https://www.candcsolicitors.co.uk/wp-content/uploads/2015/05/KidsClub7.pdf> (дата обр. 26.04.2024).
- [28] Tensorflow. *TensorFlow 2 Detection Model Zoo*. URL: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md (дата обр. 26.04.2024).
- [29] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang и Jinqiao Wang. *Fast Segment Anything*. Под ред. arXiv.org. 2001. Апр. 2023. URL: <https://arxiv.org/pdf/2306.12156> (дата обр. 26.04.2024).