Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»
Физтех-школа Прикладной Математики и Информатики
Научно-образовательный центр когнитивного моделирования

Направление подготовки / специальность: 09.04.01 Информатика и вычислительная техника
Направленность (профиль) подготовки: Методы и технологии искусственного интеллекта

ПРИМЕНЕНИЕ ГЕНЕРАТИВНЫХ МОДЕЛЕЙ ДЛЯ АУГМЕНТАЦИИ ОБУЧАЮЩИХ ДАННЫХ В ЗАДАЧАХ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ

(магистерская диссертация)

Студент:
Изместьева Ульяна Андреевна

_____
(подпись студента)

Научный руководитель:
Юдин Дмитрий Александрович,
канд. техн. наук

_____
(подпись научного руководителя)

Консультант *(при наличии)*:
_____
_____
(подпись консультанта)

Москва 2025

Содержание

1 Введение **4**

2 Анализ современных методов генерации изображений и постановка задачи **7**

2.1 Основные понятия и определения 7

2.2 Обзор современного состояния проблемы 8

2.3 Формальная постановка задачи 13

3 Разработка предлагаемого метода **15**

3.1 Теория диффузионных моделей 15

3.2 Предлагаемая функция потерь 22

3.3 Алгоритмы генерации объектов 24

3.4 Алгоритм генерации полностью синтетических данных 28

4 Вычислительные эксперименты **30**

4.1 Постановка экспериментов 30

4.2 Эксперименты генерации объектов 30

4.3 Эксперименты генерации сцен 43

4.4 Генерация синтетического датасета 44

4.5 Стек технологий 44

4.6 Выводы 45

5 Заключение **46**

Аннотация

С развитием диффузионных моделей открываются уникальные возможности для применения инновационных методов, в частности для создания синтетических обучающих наборов данных. Однако существует ряд трудностей, связанных с вычислительными ресурсами и временем, при автоматизации пополнения больших объемов данных. В данной работе представлен новый подход к генерации обучающих данных в домене, аналогичному существующим наборам данных. Предлагаемый метод основан на диффузионной модели Stable Diffusion v2, которая была дополнительно обучена с помощью **новой функции потерь** на специально собранном наборе данных TAOMR в области уличных сцен. Также реализован **алгоритм генерации полностью синтетических наборов данных**, включая синтетическую модель глубины Depth Anything v2. Код реализованного метода находится в открытом доступе: `https://github.com/Areson251/GenLab`

1 Введение

Современные методы машинного обучения требуют значительных объемов высококачественных данных для эффективного обучения моделей. Однако сбор, очистка и разбиение реальных данных зачастую требуют больших затрат времени и средств, а также подвержены человеческому фактору. В связи с этим особую актуальность приобретает задача автоматического расширения обучающих выборок за счет синтетических данных. Одним из таких перспективных направлений является использование диффузионных моделей для генерации синтетических обучающих наборов данных и улучшения процесса обучения моделей машинного обучения.

Одним из вариантов использования диффузионных моделей для аугметации датасетов является генерация синтетических объектов на изображениях в режиме inpainting, который позволяет эффективно заполнять пропущенные участки изображений, сохраняя при этом естественный вид и контекст изображения. Однако, базовых весов, лежащих в общем доступе, может быть недостаточно для ряда специфических задач (например, домен бездорожья). Появляется потребность в дообучении моделей на узкий домен.

Более того, дообучение диффузионных моделей требует большого количества размеченных данных. Это может быть проблематично в случае, когда в распоряжении ограниченное количество данных или когда процесс разметки данных является дорогостоящим и трудоемким. Также, для дообучения диффузионных моделей необходимы значительные вычислительные ресурсы, включая графические процессоры (GPU) или тензорные процессоры (TPU), а также большие объемы оперативной памяти. Это может быть ограничивающим фактором для многих исследовательских групп или компаний с ограниченными ресурсами.

Ключевыми причинами разработки инструмента автоматического расширения данных являются: необходимость повышения робастности моделей за счет разнообразных и сбалансированных обучающих выборок; решение задач в условиях ограниченного объема данных, характерных для узкоспециализированных сценариев; снижение стоимости сбора и разбиения данных за счет использования генеративных моделей, что также снижает влияние человеческого фактора.

Целью данной работы является разработка инструмента для автоматической аугментации обучающих данных на основе генеративной модели, способной создавать синтетические данные в заданном домене используя реальные данные и раз-

личные модальности.

Основания для разработки предлагаемого инструмента:

- **Расширение данных.** Наборы разнообразных данных повышают устойчивость моделей при обучении.
- **Узкоспециализированные сценарии.** Моделирование специфичных ситуаций помогает решать задачи с ограничениями данных.
- **Экономия ресурсов.** Ручной сбор и разметка данных дорогостоящая и подвержена человеческим ошибкам и значительным временным затратам.

Предлагаемый автоматический метод аугментации обучающих данных предназначен для улучшения процесса обучения моделей компьютерного зрения. Особенностью предложенного подхода является его способность выполнять преобразования не на уровне отдельных изображений, а на уровне целых наборов данных, что позволяет сохранять внутренние статистические свойства и распределения, характерные для исходной выборки. Метод полностью автономен и не требует вмешательства со стороны эксперта или дополнительной разметки, используя лишь уже существующую аннотацию, доступную в датасете. Это делает его применимым к широкому классу задач, где имеются размеченные данные, обеспечивая при этом устойчивое увеличение разнообразия обучающей выборки и потенциальное повышение обобщающей способности обучаемой модели.

В данной работе используются следующие термины:

- **Задача inpainting** - процесс восстановления или восполнения участков изображения по выделенной маске;
- **Синтетические данные** - нереальные данные, которые были получены с помощью определенного алгоритма;
- **Функция потерь** - математическая функция, которая измеряет степень расхождения между предсказанными значениями модели и истинными значениями.

Вклад представленной работы можно резюмировать следующим образом:

- Определяется новая функция потерь для дообучения диффузионных моделей с помощью LoRA[18] для генерации объектов.

- Разработан метод 12 автоматического аугментирования изображений.
- Разработан модульный нейросетевой метод 11, который генерирует новый набор данных на основе эталонных данных.
- Проведен бенчмаркинг моделей синтеза глубины.
- Расширен набор данных InstSnowRoadDetection[22] с помощью представленного метода.

2 Анализ современных методов генерации изображений и постановка задачи

Данный раздел посвящен обзору релевантных методов генерации изображений, моделей вычисления синтетической глубины и методов дообучения нейросетевых моделей. Будут введены необходимые понятия, которые в дальнейшем используются в работе, а так же определена постановка задачи и ее математическая формулировка.

Также, в данном обзоре представлен анализ разнообразных методов и подходов, направленных на решение задачи автоматизированной разметки.

2.1 Основные понятия и определения

Введем понятия и обозначения, которые будут использоваться в дальнейшем в работе:

- **Нейронная сеть** - вычислительная модель, состоящая из взаимосвязанных нейронов, организованных в слои, которая обучается на данных для решения задач
- **Обучающий набор данных** - набор репрезентативных изображений и аннотаций к ним, разделенный на обучающую, валидационную и тестовые выборки, который используется для обучения нейронной сети;
- **Латентная диффузионная модель** - тип генеративной модели, которая выполняет диффузионный процесс в латентном пространстве;
- **Задача inpainting** - процесс восстановления или восполнения участков изображения по выделенной маске;
- **Синтетические данные** - нереальные данные, которые были получены с помощью определенного алгоритма;
- **Функция потерь** - математическая функция, которая измеряет степень расхождения между предсказанными значениями модели и истинными значениями.
- **Grounded truth данные** - эталонные данные, используемые для проверки и обучения моделей;

- **Detected truth данные** - результаты, полученные с помощью алгоритмов или моделей при обработке входных данных;
- **Бенчмаркинг** - результаты сравнения различных моделей или алгоритмов;
- **Пайплайн** - последовательность автоматизированных шагов для обработки данных или выполнения задач;

2.2 Обзор современного состояния проблемы

Диффузионные модели. Первым рассмотрим модель Kandinsky [25]. В ней используется CLIP [11] в качестве энкодера текста и изображений, а также модель image prior для сопоставления изображений между латентными пространствами модальностей CLIP. Этот подход повышает визуальную производительность модели. В самой модели латентной дифуззии используют два текстовых энкодера, результаты которых потом конкаенируют. Особенность модели заключается в обучении механизма обратной диффузии восстанавливать латентное представление изображения не только из текстового эмбеддинга, а еще и из визуального эмбеддинга CLIP с условием (condition) на этот текстовый эмбеддинг. Архитектура Kandinsky содержит 3.3 миллиарда параметров.

Imagen [19] от Google состоит из замороженных весов текстового енкодера, который преобразует текст в последовательность эмбеддингов, и каскада условных диффузионных моделей, которые отображают эти эмбеддинги на изображения, постепенно улучшая их разрешение.

Альтернативная диффузионная модель Dall-E 2 [16] от OpenAI - это версия GPT-3 с 12 миллиардами параметров, обученная генерировать изображения по текстовым описаниям, используя набор данных пар "текст-изображение". DALL-E способна создавать правдоподобные образы для множества запросов, которые исследуют композиционную структуру языка.

Завершает данную часть обзора модель Stable Diffusion [17] от StabilityAI второй версии. В backbone модели лежит UNet [7]. Также в качестве тесктового энкодера используется OpenCLIP [13]. Особенность заключается в реализации автоэнкодера с *downsampling-factor*, который отвечает за понижение размерности изображений в латентном пространстве. Модели сравнительно легкая: содержит 865 миллионов парамметров.

Методы дообучения диффузионных моделей. Базовых весов, лежащих в общем доступе, может быть недостаточно для ряда специфических задач (например, домен бездорожья). Появляется потребность в дообучении моделей на узкий домен. Для решения подобных проблем были придуманы различные методы настройки моделей, не требующие прямого воздействия на их веса. Одним из таких методов является текстовая инверсия (textual inversion) [15] - техника персонализации диффузионных моделей в режиме text-to-image генерации, идея которой заключается в изменении пространства эмбеддингов в качестве цели для инверсии. Используется несколько изображений для обучения текстовой модели новому слову, обозначаемое как $S*$, и дообучению векторных представлений $v*$, близких к некоторому визуальному представлению. Это достигается путем добавления новой лексемы в словарный запас и обучения на нескольких репрезентативных изображениях. Таким образом, можно запросить "фотографию $S*$"или "рисунок в стиле $S*$". Этот процесс не вносит никаких изменений в веса генеративной модели, поэтому не требуется большое количество времени и других ресурсов для его обучения.

Альтернативный вариант кастомизации моделей является метод DreamBooth [23]. По $3-5$ изображениям объекта, авторы настраивают модель диффузии text2image, используя входные изображения в паре с текстовой подсказкой, содержащей уникальный идентификатор и название класса, к которому принадлежит объект (например, "A [V] собака"), параллельно применяея функцию потерь сохранения приоритета (class-specific prior preservation loss) для специфичного класса, которая использует семантический приоритет, имеющийся у модели, и побуждает ее генерировать разнообразные экземпляры, принадлежащие к классу объекта, используя название класса в текстовой подсказке (например, "A собака").

Custom Diffusion [26] отличается от предыдущих рассмотренных методов тем, что именно в его процессе происходит непосредственное влияние на веса генеративных моделей. Так же, используя $4-5$ изображений, представленная техника дообучает веса в слоях перекрестного внимания (cross-attention layers), и использует специальное слово для представления новой выученной концепции. Уникальность Custom Diffusion заключается в том, что она может одновременно изучать несколько концепций.

И завершает данную секцию метод LoRA[18]. Его основная идея заключается в работе с рангом более низких размерностей. Если взять матрицу предобученной и обученной модели, вычесть друг из друга и посчитать ранг, то он окажется меньше

чем ее размерность. Также, параллельно слоям добавляем два слоя адаптера, где первый слой даунсемплит в скрытое пространство, затем второй адаптер преобразует в исходную размерность, после чего складываются выходы. Плюсы данного метода заключаются в возможности относительно "дешего"дообучить большую модель под свои данные.

Модели выччисления синтетической глубины. Открывает данный подраздел модель Depth Anything[29], которая основана на сверточной нейронной сети. Обучается модель на огромном наборе данных (1.5M+ изображений). Быстрая и точная, подходит для реального времени. Имеет хороший баланс скорости и качества и поддерживает различные сценарии. Однако, эта модель очень чувствительна к текстурам и освещению.

Ее продолжение Depth Anything v2 более точно работает с мелкими деталями и сложными сценами. Имеет улучшенную аугментацию данных и архитектуру, а так же лучше справляется с occlusion и сложными текстурами.

Модель MaryGold[31] уже использует гибрид сверточной нейронной сети и трансформенной архитектуры. Хорошо работает в условиях слабой освещенности, но, к сожалению, отсутствует документация в открытом доступе.

DPT[14] является Vision Transformer (ViT) для предсказания глубины. Славится высокой точностью, особенно на границах объектов. Но при этом большом плюсе имеет высокие требования к GPU и значительно медленнее CNN-аналогов.

И завершает обзор моделей вычисления синтетической глубины модель DepthPRO[30], которая является коммерческим проектом NVIDIA. Она оптимизирована для промышленного применения (робототехника, автономные системы), но является закрытой для общего доступа моделью.

Модели сегментации, совмещающие модальности текста и изображений. Открывает обзор задача поиска месторасположения искомых объектов на входных изображениях и текстовых описаниях. Формулировка задачи звучит следующим образом: дан набор изображений, их разметка в формате, указывающем положение и/или контуры объектов на изображениях, и связанный с ними текстовый контент. Требуется разработать мультимодельную архитектуру, способную обрабатывать и интегрировать информацию из изображений и текста для выполнения задач анализа и предсказания.

Первый метод MDETR [12] демонстрирует разрабатку *конечного (end-to-end) модулированного детектора*, способного обнаруживать объекты на изображениях в зависимости от сырых текстовых запросов. Это позволяет достичь беспрепятственного объединения текстовых данных и изображений, обеспечивая мультимодальное рассуждение. MDETR основан на фреймворке DETR [10] и выполняет обнаружение объектов совместно с пониманием естественного языка, обеспечивая истинно конечное мультимодальное рассуждение. MDETR полностью полагается на текст и согласованные рамки для наблюдения и может обнаруживать тонкие концепции из свободного текста, обобщаясь на невидимые комбинации. Он достигает передовых результатов благодаря предварительному обучению и донастройке.

Представленный подход обладает следующими особенностями:

- обнаружение тонких концепций: MDETR способен обнаруживать тонкие концепции из свободного текста и обобщать их до невидимых комбинаций категорий и атрибутов. Эта гибкость в обнаружении концепций выделяет его среди существующих методов обнаружения.
- полная дифференцируемость: MDETR полностью дифференцируем.

Целью Grounding DINO [24] является расширение возможностей существующих моделей для поддержки обнаружения объектов с использованием языковых запросов и достижения высокой производительности на нулевой передаче знаний (zero-shot). Авторы рассматривают новую модель обнаружения объектов с *открытым множеством классов* (open-set) под названием Grounding DINO. Представленная модель расширяет закрытый детектор DINO для выполнения обнаружения объектов с использованием языковых запросов благодаря предварительному обучению *видеоязыковых моделей* (vision-language pre-training). Авторы вводят текстовые особенности на нескольких этапах, включая кодировщик особенностей, выбор запросов под руководством языка и кросс-модальный декодер. Grounding DINO демонстрирует высокую производительность на нулевой передаче знаний (zero-shot) на популярных бенчмарках, таких как COCO [6], LVIS [9] и ODinW .

В итоге, Grounding DINO устанавливает новую парадигму для обнаружения объектов с открытым множеством классов через свою объединенную трансформаторную архитектуру и многофазный подход к объединению данных из зрительной области и языка.

Также, с текстовыи запросами работает модель, представленная в статье [20]. Авторы предлагают простой и эффективный метод передачи предварительно обученных моделей изображений и текста (image-text models) с контрастным обучением для выполнения задачи обнаружения объектов с *открытым словарем* (open-vocabulary). В ходе обширных экспериментов анализируется влияние различных факторов, таких как размер модели, длительность предварительного обучения, аугментация данных и другие, на конечную производительность обнаружения. Предложенный метод достигает результатов на уровне передовых технологий как на *нулевых условиях текстово-условного обнаружения* (zero-shot text-conditioned), так и на условиях *одного образца изображенно-условного обнаружения* (one-shot image-conditioned).

Кроме того, был проанализирован фреймворк, продемонстрированный в работе [21]. Авторы решают задачу разработки систем зрения, способных быть применимыми к новым концепциям или областям, особенно уделяя внимание открытой вокабулярной детекции объектов и семантической сегментации. Они стремятся преодолеть разногласия между задачами детекции и сегментации, используя большие наборы изображений с текстовой информацией для более точной работы систем зрения. Предложенная модель OpenSeeD нацелена на совместное обучение на данных детекции и сегментации для создания единой модели с открытым словарем для обеих задач, учитывая различия в размерах словарей и пространственной детализации наблюдений. OpenSeeD - это простой фреймворк для обнаружения объектов с открытым словарем и семантической сегментацией. Он использует наборы данных обнаружения с прямоугольными и масочными аннотациями для совместного обучения одной модели. Для учета различий в задачах и данных он использует *декодирование переднего и заднего плана* (foreground/background), а также условное декодирование маски, которое генерирует псевдо-маски для данных обнаружения. Он достигает лучших результатов при переносе обучения на конкретные задачи.

Выводы. Проанализировав широкораспространенные диффузионные модели и методы их дообучения за основу в экспериментах была выбрана Stable Diffusion, так как она имеет достаточно простую и понятную архитектуру, весит сравнительно меньше других рассмотренных моделей, а также открытый код. В качестве инструмента для обучения выбран метод LoRA по причине возможности обобщения моделей на большое количество генерируемых объектов, поддержке обработки большого количества изображений и легкой имплементации новых блоков кода в существующий

в открытом доступе.

Для тестирования аугментированных датасетов была выбрана модель сегментации YOLOv8 medium, так как данная модель не требует большого количества обучающих данных, быстро обучается и имеет низкое время инференса.

2.3 Формальная постановка задачи

Поставленную задачу необходимо разделить на два пункта:

1. Разработать модель F, которая по входным данным X генерирует выходные данные Y. F: $X \rightarrow Y$, где:

   - **Входные данные X:** набор размеченных изображений *(I)*; маски объектов *(M)*; текстовые описания объектов (категории классов) *(T)*;
   - **Выходные данные Y**: аугментированные изображения *(I')*; аугментированный список масок с новыми объектами *(M')*; текстовые описания объектов (категории классов) *(T)*; аугментированные карты глубин *(D')*;

2. Разработать алгоритм A: $Y \rightarrow Z$ преобразования входных данных Y в выходные данные Z, где:

   - **Входные данные Y:** набор аугментированных изображений *(I')*; маски объектов *(M')*; текстовые описания объектов (категории классов) *(T)*; аугментированные карты глубин *(D')*;
   - **Выходные данные Z:** аугментированные изображения *(I")*; аугментированный список масок с новыми объектами *(M")*; текстовые описания объектов (категории классов) *(T)*; аугментированные карты глубин *(D")*;

Математическая формулировка поставленной задачи выглядит следующим образом:

Пусть $G_\theta(\hat{c})$ - это модель, которая отображает условие класса объекта в виде текста маски объекта $\hat{c}$ в условный вектор. Тогда, задача поиска функции F сводится к задаче оптимизации функции потерь $L_{inpaint}$ вида:

$$\min_\theta L_{inpaint} := \mathbb{E}_{\epsilon \sim \mathrm{N}(0,1), z, t, \hat{c}}[\|\epsilon - \epsilon_\theta(z_t, t, G_\theta(\hat{c}))\|^2] \tag{1}$$

где:

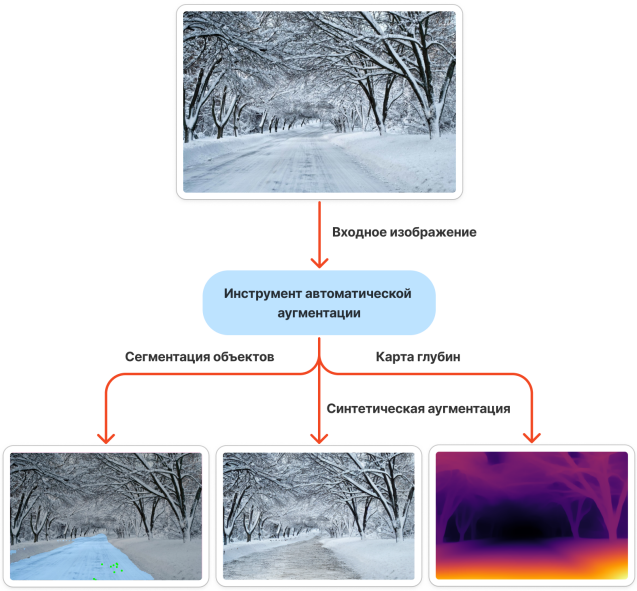

Рис. 1: Визуальное представление постановки задачи.

- t - шаг во времени;
- z_t - латентное представление изображения, зашумленное в момент времени t;
- ϵ - образец немасштабированного шума;
- θ - нейронная сеть расшумления.

3 Разработка предлагаемого метода

В данной секции будет рассказанно подробно о постановке идеи нового метода обучения диффузионных моделей за счет изменения функции потерь. Также, будут описаны алгоритмы генерации синтетических данных.

3.1 Теория диффузионных моделей

Идея диффузионных моделей пришла из неравновесной термодинамики. По Марковской цепочке определяется каждый шаг диффузии по добавлению случайного шума в семпл данных, а затем они учатся пошагово убирать этот шум для реконструкции входных данных. Основное отличие диффузионных моделей от GAN и VAE заключается в том, что они работают в латентном пространстве той же размерности. Из наблюдаемых семплов x из интересующего нас распределения цель генеративной модели - научиться моделировать истинное распределение данных $p(x)$. После обучения мы можем произвольно генерировать новые семплы нашей моделью аппроксимации.

В контексте генеративных моделей это означает, что модель обучается так, чтобы вероятность наблюдаемых данных была максимальной. В частности, если у нас есть набор данных $X = \{x_1, x_2, ..., x_N\}$, то задача заключается в нахождении параметров модели, которые максимизируют правдоподобие данных $P(X)$.

Для сложных моделей, где данные зависят от скрытых переменных z полное правдоподобие $P(X)$ можно выразить через интеграл по скрытым переменным:

$$P(X) = \int P(X, z)\, dz = \int P(X|z)P(z)\, dz$$

Однако этот интеграл может быть вычислительно сложным, и вычисление $P(X)$ напрямую часто невозможно.

Для этого используется **ELBO**, который является нижней границей на логарифм правдоподобия:

$$\log P(X) \geqslant \mathbb{E}_{q_\varphi(z|X)}[\log P(X, z)] - \mathbb{E}_{q_\varphi(z)}[\log q_\varphi(z|X)]$$

Здесь $q_\varphi(z|X)$ — это вариационное распределение, которое аппроксимирует истинное распределение скрытых переменных $P(z|X)$. ELBO — это нижняя граница на логарифм правдоподобия, и её максимизация эквивалентна максимизации правдоподобия данных, но с использованием аппроксимации через $q_\varphi(z|X)$.

ELBO разлагается следующим образом:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_q \left[\sum_{t=1}^{T} D_{\text{KL}}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t)) - \log p_\theta(x_0|x_1) \right].$$

1. КЛ-дивергенция между $q(x_{t-1}|x_t, x_0)$ и $p_\theta(x_{t-1}|x_t)$:
   - Эта часть измеряет, насколько хорошо модель p_θ аппроксимирует истинное распределение обратного процесса q.
   - Минимизация этой дивергенции является основной задачей обучения.
2. Логарифм правдоподобия данных $\log p_\theta(x_0|x_1)$:
   - Эта часть отвечает за финальное восстановление данных из латентного состояния x_1.

Упрощение задачи обучения:

1. Аппроксимация $q(x_{t-1}|x_t, x_0)$:
   (a) $q(x_{t-1}|x_t, x_0)$ представляется в аналитической форме как гауссовское распределение.
   (b) Это упрощает вычисление КЛ-дивергенции.
2. Предсказание шума:
   - Вместо предсказания параметров x_{t-1}, модель обучается предсказывать добавленный шум $\epsilon_\theta(x_t, t)$, что делает задачу более стабильной.

Итоговая функция потерь ELBO приводит к функции потерь, которая сводится к минимизации среднеквадратичной ошибки (MSE) между истинным шумом ϵ и предсказанным шумом $\epsilon_\theta(x_t, t)$:

$$\mathcal{L}_t = \mathbb{E}_{x_0, \epsilon, t} \left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2 \right].$$

Эта формула используется на практике, так как она упрощает обучение, сохраняя при этом связь с теоретической основой ELBO.

ELBO играет ключевую роль в теории диффузионных моделей, так как:

1. Формализует задачу максимизации правдоподобия данных;
2. Делает обучение модели статистически обоснованным;

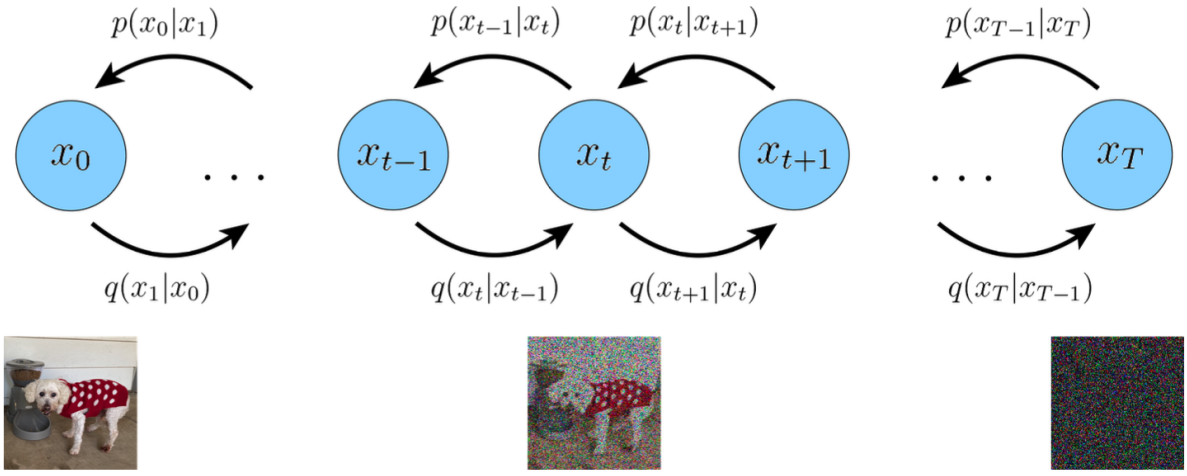

Рис. 2: Схема прямого процесса диффузии.

3. Обеспечивает разложение задачи на понятные компоненты (например, минимизацию КЛ-дивергенции).

Прямой процесс диффузии Дан семпл из распределения данных $\mathbf{x}_0 \sim q(\mathbf{x})$. Один шаг диффузии - добавление небольшого количества нормального шума на T-ом шаге. Каждый шаг диффузии порождает последовательность зашумленных примеров $\mathbf{x}_1, \ldots, \mathbf{x}_T$. Размеры шагов регулируются графиком отклонений $\{\beta_t \in (0,1)\}_{t=1}^T$. Схема прямого процесса дифузии представлена на рисунке 2.

$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$ $q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})$

Эта формула описывает **переходное распределение** $q(\mathbf{x}_t|\mathbf{x}_{t-1})$.

1. $q(\mathbf{x}_t|\mathbf{x}_{t-1})$: Это условное распределение переменной $\mathbf{x}_t$ на текущем шаге t, если известно значение $\mathbf{x}_{t-1}$ на предыдущем шаге.

2. $\mathcal{N}(\cdot; \mu, \Sigma)$: Это обозначение для многомерного нормального распределения с:

   - математическим ожиданием μ,
   - ковариационной матрицей Σ.

3. Математическое ожидание (μ):

   - $\sqrt{1-\beta_t}\mathbf{x}_{t-1}$: Это сдвиг, определяющий, как среднее значение $\mathbf{x}_t$ зависит от предыдущего состояния $\mathbf{x}_{t-1}$. Коэффициент $\sqrt{1-\beta_t}$ уменьшает вклад $\mathbf{x}_{t-1}$, постепенно снижая его влияние с ростом β_t.

4. Ковариационная матрица (Σ):

- $\beta_t\mathbf{I}$: Это изотропный шум (шум одинаковой величины во всех направлениях), где β_t определяет величину шума. $\mathbf{I}$ — единичная матрица, которая делает шум независимым по всем координатам.

Смысл заключается в:

- Добавлении шума: Формула описывает процесс добавления шума на каждом шаге t. Чем больше β_t, тем больше шума добавляется на текущем шаге.
- Контроле шума: Параметр β_t регулирует, насколько сильно текущее состояние $\mathbf{x}_t$ отклоняется от предыдущего $\mathbf{x}_{t-1}$ из-за шума.

Если β_t увеличивается с течением времени, то процесс становится всё более шумным. В конечном итоге $\mathbf{x}_t$ становится почти полностью случайным шумом. В обратном процессе (восстановлении) модель обучается убирать этот шум и восстанавливать исходные данные.

Особенность прямого процесса заключается в том, что нам не нужно вычислять итеративно каждый шаг зашумления - мы можем сразу определить $\mathbf{x}_t$ на шаге t с помощью [reparameterization trick](https://lilianweng.github.io/posts/2018-08-12-vae/reparameterization-trick). The reparameterization trick позволяет переписать случайную переменную как детерминированную функцию шумовой переменной - это позволяет оптимизировать нестохастичные члены с помощью градиентного спуска. Например, выборки из нормального распределения $xN(x; \mu, \sigma^2)$ с произвольным средним μ и дисперсией σ^2 могут быть переписаны в виде:

$$x = \mu + \sigma\varepsilon \quad \varepsilon N(\varepsilon; 0, I)$$

Пусть $\alpha_t = 1 - \beta_t$ и $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. Тогда:

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} && \text{;где } \boldsymbol{\epsilon}_{t-1}, \boldsymbol{\epsilon}_{t-2}, \cdots \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\boldsymbol{\epsilon}}_{t-2} && \text{;где } \bar{\boldsymbol{\epsilon}}_{t-2} \text{ объединяет два гауссиана (*).} \\
&= \dots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \\
q(\mathbf{x}_t|\mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})
\end{aligned}$$

$(*)$ При объединении двух гауссианов с разными дисперсиями, $\mathcal{N}(\mathbf{0}, \sigma_1^2\mathbf{I})$ и $\mathcal{N}(\mathbf{0}, \sigma_2^2\mathbf{I})$, то новое распределение будет $\mathcal{N}(\mathbf{0}, (\sigma_1^2+\sigma_2^2)\mathbf{I})$. Здесь объединенное стандартное отклонение $\sqrt{(1-\alpha_t) + \alpha_t(1-\alpha_{t-1})} = \sqrt{1-\alpha_t\alpha_{t-1}}$. Обычно с каждым шагом зашумления $\beta_{t-1} < \beta_t$.

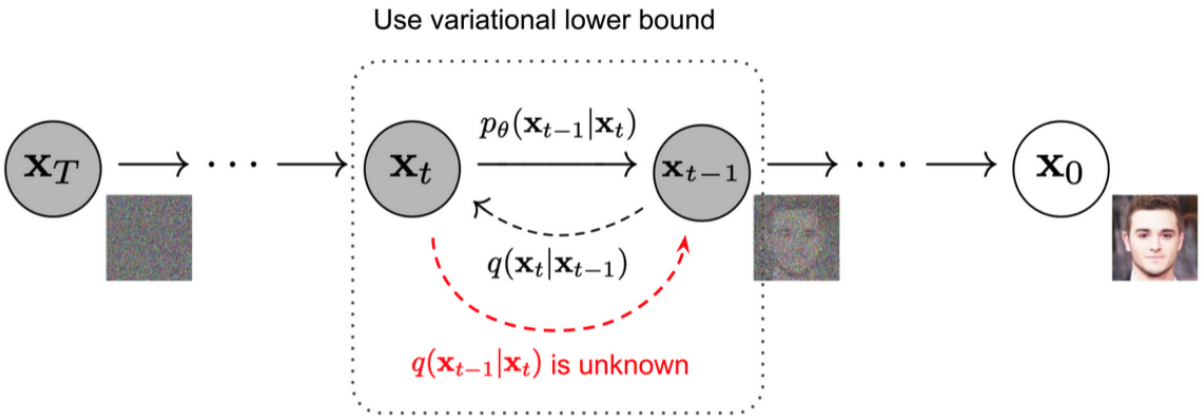

Рис. 3: Схема прямого процесса диффузии.

Связь со стохастической градиентной динамикой Ланжевена Динамика Ланжевена - это концепция из физики, разработанная для статистического моделирования молекулярных систем. В сочетании со стохастическим градиентным спуском можно производить выборки из плотности вероятности $p(\mathbf{x})$, используя только градиенты $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ в марковской цепи обновлений:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \frac{\delta}{2}\nabla_{\mathbf{x}} \log p(\mathbf{x}_{t-1}) + \sqrt{\delta}\boldsymbol{\epsilon}_t, \quad \text{где } \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

где δ задает размер шага. При $T \to \infty, \epsilon \to 0, \mathbf{x}_T$ стремится к истинной плотности вероятности $p(\mathbf{x})$. Ньюанс: в отличие от классического стохастического градиентного спуска (SGD), стохастическая градиентная динамика Ланжевена включает добавление нормального шума ($\sqrt{\delta}\boldsymbol{\epsilon}_t$), чтобы избежать застревания в локальных минимумах в процессе обучения.

Обратный процесс диффузии Необходимо обернуть процесс, описанный выше, и научиться семплировать $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, чтобы восстанавливать данные из шума $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 3. К сожалению, мы не можем точно вычислять $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ (требует знания всех промежуточных состояний или параметров), поэтому нам нужна аппроксимация в виде обучаемой модели p_θ.

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

где

- $\mu_{(x_t,t)}$ — предсказанная моделью средняя (mean);
- $\Sigma_\theta(x_t, t)$ — ковариационная матрица (обычно диагональная).

Обычно $\mu_\theta(x_t, t)$ выражается через предсказание исходного изображения x_0:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right),$$

где:

- $\alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$;
- $\epsilon_\theta(x_t, t)$— предсказание добавленного шума, которое выдает модель.

Таким образом, вместо непосредственного предсказания μ_θ, модель обучается предсказывать шум ϵ_θ, что делает обучение более стабильным.

Алгоритм обратного процесса:

1. Инициализация: начать с гауссовского шума $x_T \sim \mathcal{N}(0, I)$.
2. Итеративное обновление:
   (a) На каждом шаге t, используя текущий зашумленный пример x_t , модель предсказывает шум $\epsilon_\theta(x_t, t)$.
   (b) - На основе предсказанного шума вычисляется $\mu_\theta(x_t, t)$.
   (c) Генерируется x_{t-1} путём семплирования из $\mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$.
3. Окончание: после T шагов получается x_0, соответствующий сгенерированным данным.

Условная генерация При обучении генеративных моделей на изображениях с условной информацией, таких как набор данных ImageNet, принято генерировать образцы, обусловленные метками классов или фрагментом текста описания. Classifier Guided Diffusion. Чтобы явно включить информацию о классе в процесс распространения, Dhariwal и Nichol обучили классификатор $f_\varphi(y|\mathbf{x}_t, t)$ на зашумленном изображении $\mathbf{x}_t$ и использовали градиенты $\nabla_{\mathbf{x}} \log f_\varphi(y|\mathbf{x}_t)$, чтобы направить процесс диффузионной выборки в сторону условной информации y (например, метки целевого класса) путем изменения предсказания шума. Вспомним, что $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ и мы можем записать функцию оценки для совместного распределения $q(\mathbf{x}_t, y)$ следующим образом:

$$\begin{aligned}
\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t, y) &= \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log q(y|\mathbf{x}_t) \\
&\approx -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + \nabla_{\mathbf{x}_t} \log f_\varphi(y|\mathbf{x}_t) \\
&= -\frac{1}{\sqrt{1-\bar{\alpha}_t}} (\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) - \sqrt{1-\bar{\alpha}_t} \nabla_{\mathbf{x}_t} \log f_\varphi(y|\mathbf{x}_t))
\end{aligned}$$

Таким образом, новый классификатор-предиктор $\bar{\boldsymbol{\epsilon}}_\theta$ будет иметь следующий вид:

$$\bar{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t) = \boldsymbol{\epsilon}_\theta(x_t, t) - \sqrt{1-\bar{\alpha}_t}\nabla_{\mathbf{x}_t} \log f_\varphi(y|\mathbf{x}_t)$$

Чтобы контролировать силу руководства классификатора, мы можем добавить вес w к дельта-части:

$$\bar{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t) = \boldsymbol{\epsilon}_\theta(x_t, t) - \sqrt{1-\bar{\alpha}_t}\; w\nabla_{\mathbf{x}_t} \log f_\varphi(y|\mathbf{x}_t)$$

Полученная ablated diffusion model (ADM) и additional classifier guidance (ADM-G) способны достичь лучших результатов, чем генеративные модели SOTA (например, BigGAN). ![[Screenshot 2025-01-13 at 12.09.32.png]]

Кроме того, Dhariwal и Nichol, внеся некоторые изменения в архитектуру U-Net, показали производительность выше, чем у GAN с диффузионными моделями. Модификации архитектуры включают большую глубину/ширину модели, большее количество голов внимания, multi-resolution attention, BigGAN residual blocks для повышения/понижения выборки, residual connection rescale $1/\sqrt{2}$ и адаптивную групповую нормализацию (AdaGN).

Classifier-Free Guidance. При отсутствии независимого классификатора f_φ можно выполнить условные шаги диффузии, объединив оценки условной и безусловной моделей диффузии ([Ho Salimans, 2021](https://openreview.net/forum?id=qw8AKxfYbI)). Пусть безусловная диффузионная модель расшумления $p_\theta(\mathbf{x})$ параметризована через оценку скоров $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$, а условная модель $p_\theta(\mathbf{x}|y)$ - через $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y)$. Эти две модели могут быть обучены с помощью одной нейронной сети. Точнее, условная модель диффузии $p_\theta(\mathbf{x}|y)$ обучается на парных данных $(\mathbf{x}, y)$, где условная информация y периодически случайным образом отбрасывается, чтобы модель умела генерировать изображения и в безусловном режиме, т.е. $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) = \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y = \varnothing)$. Градиент неявного классификатора может быть представлен с помощью условных и безусловных оценок. После включения в модифицированную оценку, управляемую классификатором, оценка не зависит от отдельного классификатора.

$$\begin{aligned}
\nabla_{\mathbf{x}_t} \log p(y|\mathbf{x}_t) &= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|y) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) \\
&= -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\Big(\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\Big) \\
\bar{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, y) &= \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - \sqrt{1-\bar{\alpha}_t}\; w\nabla_{\mathbf{x}_t} \log p(y|\mathbf{x}_t) \\
&= \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) + w\big(\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\big) \\
&= (w+1)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - w\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)
\end{aligned}$$

Итоговая функция потерь

$$\mathcal{L} = \mathbb{E}_{x_0,\epsilon,t}\left[\lambda \cdot \|\epsilon - \epsilon_\theta(x_t, t, y)\|^2 + (1-\lambda) \cdot \|\epsilon - \epsilon_\theta(x_t, t, \emptyset)\|^2\right],$$

где:

- λ — коэффициент, контролирующий вклад условной и безусловной частей.

3.2 Предлагаемая функция потерь

В данной работе диффузионная модель Stable Diffusion v2 была обучена с использованием новых функций потерь. Первоначальная идея **Inpainting Contrastive Loss (IC Loss)**заключается в следующем: мы дублируем и замораживаем оригинальные веса Unet, на выходе которых будет ϵ_{empt} шум. Затем, используя условный вектор, который получаем из текстового промпта, мы просим модель удалить объект с изображения, получая таким образом «пустое» изображение и подсчитывая обратную разницу между шумом ϵ_θ, предсказанным обученными весами LoRA, и шумом ϵ_{empt}, предсказанным замороженными весами. Это второй член в общей функции ошибки. Таким образом, мы не только поощряем модель генерировать объекты, но и наказываем ее, когда она пытается сгенерировать «пустоту».

$$\min_\theta L_{IC} := \alpha\|\epsilon_{obj} - \epsilon_\theta(z_t, t, G_\theta(\hat{c}))\|^2 + \frac{\beta}{\|\epsilon_{empt} - \epsilon_\theta(z_t, t, G_\theta(\hat{c}))\|^2 + \mu} \quad (3.1)$$

где:

- ϵ_{obj} - шум GT объекта;
- ϵ_{empt} - шум синтетически сгенерированной пустоты;
- $G_\theta(\hat{c})$ - выход модели кодирования условия;
- α и β - весовые коэффициенты;
- μ - обработка исключения деления на ноль.

Однако, как показали результаты в секции 4, предложенная функция потерь является несовершенной, так как модель диффузии стремится затемнить все пиксели максированного участка изображения, потому что так выгоднее получить разницу межд grounded true сгенерированной "пустотой"и генерируемым объектом.

В следствие чего предлагается использовать обновленную функцию потерь **Regularized Inpainting Contrastive Loss (RIC Loss)**, которая имеет вид:

$$\begin{aligned}\min_{\theta} L_{\text{RIC}} :=& \alpha \left\| \epsilon_{\text{obj}} - \epsilon_\theta\big(z_t, t, G_\theta(\hat{c})\big) \right\|^2 \\ &+ \frac{\beta}{\left\| \epsilon_{\text{empt}} - \epsilon_\theta\big(z_t, t, G_\theta(\hat{c})\big) \right\|^2 + \mu} \\ &+ \gamma \left\| \epsilon_{\text{unmasked}} - \epsilon_\theta\big(z_t, t, G_\theta(\hat{c})\big) \right\|^2,\end{aligned} \tag{3.2}$$

где:

- ϵ_{obj} - шум GT объекта;
- ϵ_{empt} - шум синтетически сгенерированной пустоты;
- ϵ_{empt} - шум немаскированной части изображения;
- $G_\theta(\hat{c})$ - выход модели кодирования условия;
- α, β и γ - весовые коэффициенты;
- μ - обработка исключения деления на ноль.

Особенность предложенной функции потерь заключается в **новом слагаемом:** теперь мы штрафуем модель при попытке увеличить разницу с grounded true с помощью затемнения немаскированной области изображения. Сейчас мы считаем разницу внутри немаскированной области изображения.

Также, для повышения эффективности вычислений во втором члене данной формулы предлагается считать разницу шумов только внутри маскированной области, что будет логичным при добавлении третьего члена.

Для измерения качества изображений были выбраны следующие метрики.

CLIP Score 3.3 измеряет соответствие между текстовым описанием и изображением, используя предобученную модель CLIP для получения эмбеддингов текста E_{text} и изображения E_{image}. Считается как косинусное сходство между ними, умноженное на 100. Чем выше значение, тем лучше совпадение.

$$\text{CLIP Score} = \max(100 \cdot \cos(E_{\text{text}}, E_{\text{image}}), 0) \tag{3.3}$$

SSIM 3.4 оценивает структурное сходство между двумя изображениями x и y, сравнивая яркость μ, контрастность σ и корреляцию σ_{xy}. Константы $C1$,$C2$ добавляются для стабильности вычислений. Значение варьируется от -1 до 1; чем ближе к 1, тем более похожи изображения.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3.4}$$

PSNR 3.5 измеряет качество реконструкции изображения через среднеквадратичную ошибку (MSE). MAX_I — максимальное возможное значение пикселя (например, 255 для 8-битного изображения). Высокий PSNR указывает на малую ошибку, но не всегда коррелирует с восприятием человеком.

$$\text{PSNR}(x, y) = 10 \cdot \log_{10}\left(\frac{\text{MAX}_I^2}{\text{MSE}(x, y)}\right), \tag{3.5}$$

где $\text{MSE}(x, y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2$

FID 3.6 сравнивает распределения признаков реальных r и сгенерированных g изображений, полученных с помощью Inception-v3. Здесь — средние значения, — матрицы ковариации. Меньшее значение FID означает лучшее качество генерации.

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}) \tag{3.6}$$

LPIPS 3.7 измеряет перцептуальное различие между изображениями, используя предобученные CNN (например, VGG или AlexNet). Для каждого слоя сети l вычисляется взвешенная евклидова разность между нормализованными фичами $\hat{f}_l$. Малое значение LPIPS указывает на высокое перцептуальное сходство.

$$\text{LPIPS}(x, y) = \sum_l \frac{1}{C_l H_l W_l} \sum_{i,j,k} \|w_l \circ (\hat{f}_{ijk}^l(x) - \hat{f}_{ijk}^l(y))\|_2^2 \tag{3.7}$$

3.3 Алгоритмы генерации объектов

Для экспериментов были выбраны две модели латентной диффузии: Stable Diffusion 4 и PowerPaint 5. Отличие этих моделей заключается в том, что авторы модели PowerPaint взяли за основу так же Stable Diffusion и дообучили их особым образом: использовали несколько видов промптов, описывающих контекст общей сцены на изображении, генерируемый объект и отсутствие объека на изображении (негативный пример). Для тюнинга text2image используют обучаемые промпты P_{obj} и P_{ctxt}. Для P_{obj} используются bbox как маски объекта, а для P_{ctxt} используется описание картинки. Используют P_{obj} как негативный промпт и P_{ctxt} как позитивный для задачи удаления объекта. Обучают P_{shape} на сегментационных масках объектах его описания. Данная модель казалась подходящей для решения поставленной задачи,

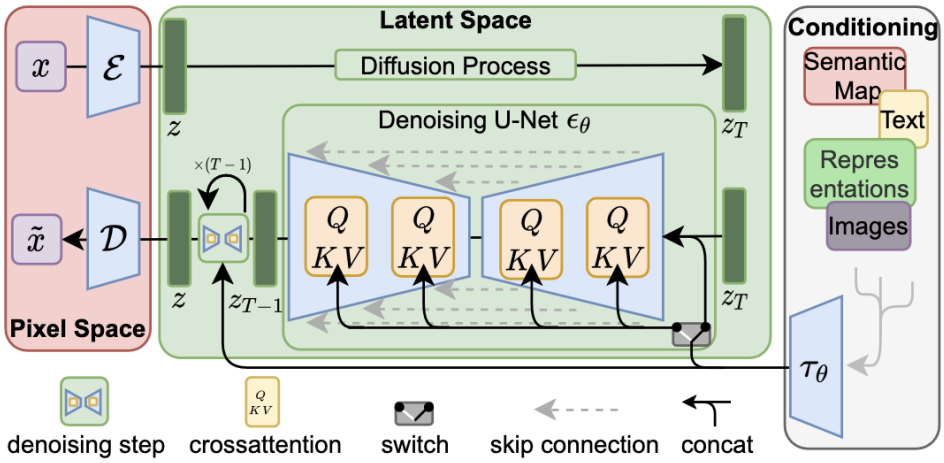

Рис. 4: Схема модели Stable Diffusion.

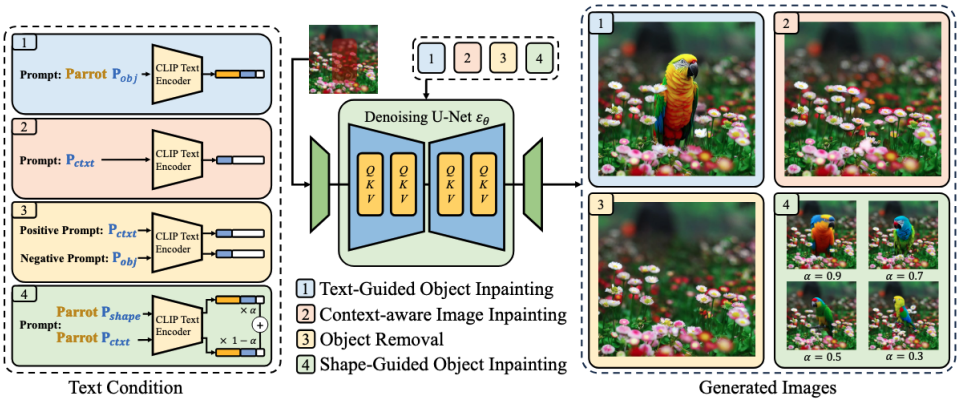

Рис. 5: Схема модели PowerPaint.

поэтому было принято решение так же ее протестировать. Результаты ее тестирования приведены в секции экспериментов.

Как и говорилось ранее, текстовая инверсия работает за счет подбора оптимальных значений для эмбеддинга $v*$ для заданного слова $S*$ 6.

Соответственно, задача заключается в замене текстового эмбеддинга $v*$ на векторное представление, отвечающее за представления не только текста, но и масок объектов - бинарных изображений 7.

Также, было принято решение протестировать алгоритм DreamBooth для поставленной задачи. Метод позволяет персонализировать вывод, обучая модель на нескольких изображениях конкретного объекта или стиля. В отличие от текстовой инверсии, DreamBooth связывает уникальный токен (например, "sks") с объектом, сохраняя его визуальные особенности 8. Модель дообучается на этих изображениях, сохраняя общие знания из исходного датасета, но теперь может генерировать объект в новых контекстах по запросу (например, "sks dog in a park"). Это достигается через

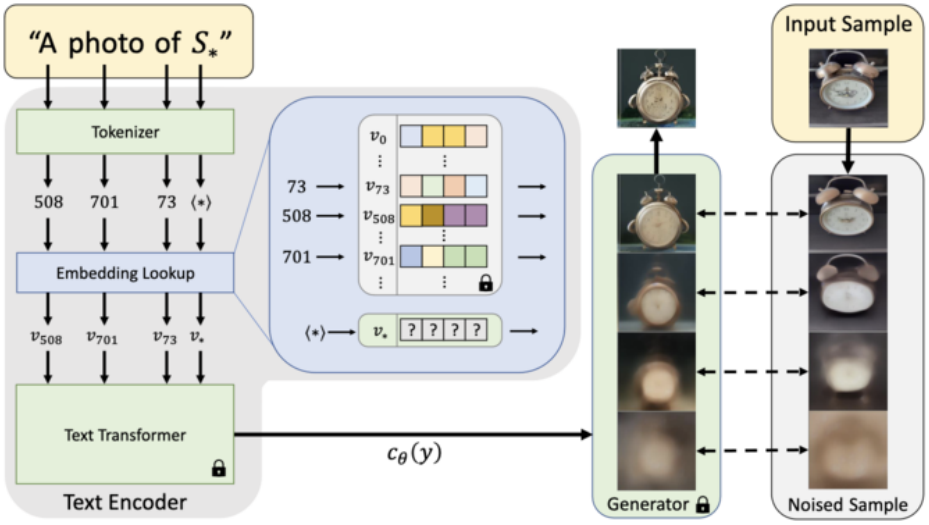

Рис. 6: Схема алгоритма текстовой инверсии.

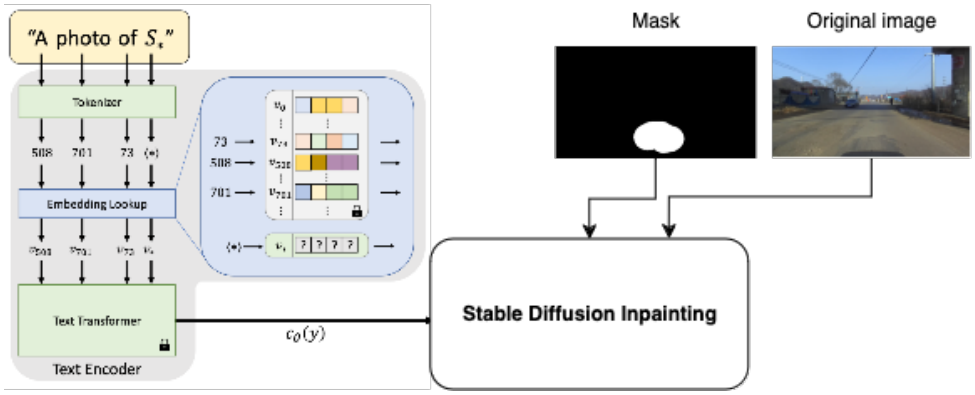

Рис. 7: Схема модифицированного алгоритма текстовой инверсии.

контролируемое обучение с сохранением семантической согласованности.

Однако, в открытом доступе, как и с текстовой инверсией, был код только для text2image задачи. Поэтому данный алгоритм был так же модифицирован под задачу inpainting 9.

Рассмотренные методы выше работают только с одинм классом объекта и дообучают модель на одно его представление (требуется всего 3-5 изображений). Для задачи аугментации хотелось бы иметь возможность добавлять большое количество классов и при этом иметь достаточную вариативность в их генерации. Для этого подходит распространенный метод дообучения LoRA 10.

Для его использования не нужна была адаптация под задачу inpainting, для этого необходимо было изменить только вход модели расшумления Unet - добавить каналы для маски изображения - и переписать процесс загрузки датасета для про-

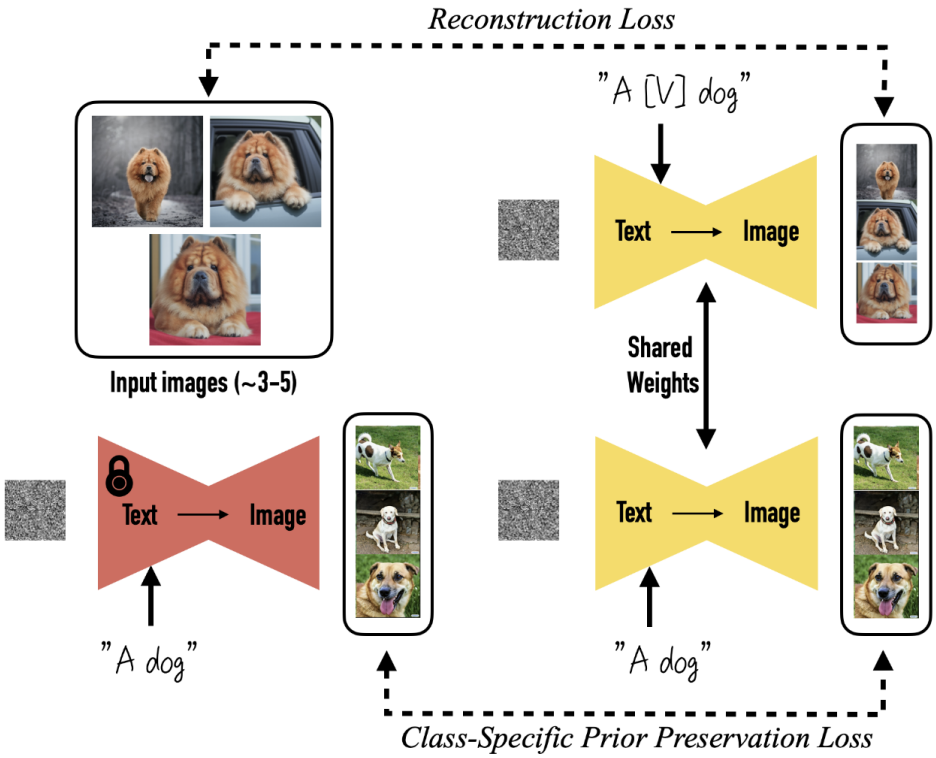

Рис. 8: Схема алгоритма DreamBooth.

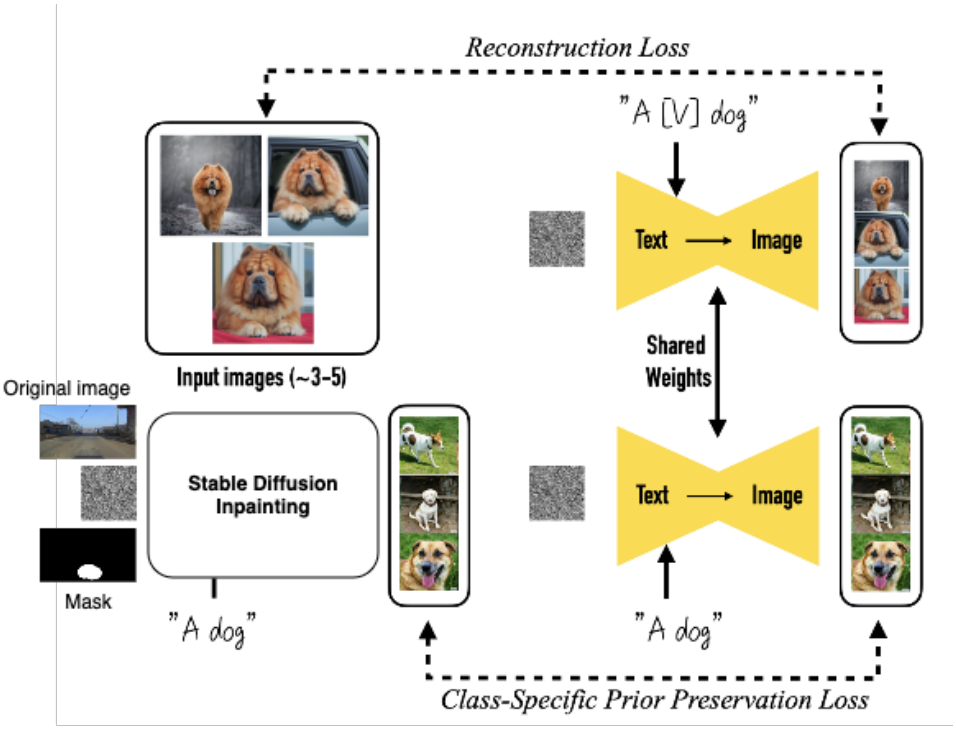

Рис. 9: Схема модифицированного алгоритма DreamBooth.

цесса обучения.

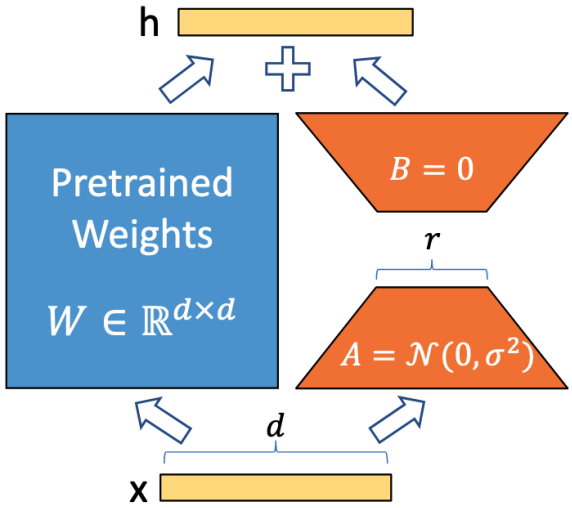

Рис. 10: Схема алгоритма LoRA.



Рис. 11: Схема алгоритма генерации полностью синтетических данных.

3.4 Алгоритм генерации полностью синтетических данных

Ниже представлена схема 11 мультимодельного диффузионного пайплайна для дополнения данных визуальной и сегментационной семантикой. Сначала из входного изображения вычисляется синтетическая глубина. На основе этой глубины генерируется изображение сцены. Размер объекта в маске генерации входной цели рассчитывается с использованием глубины в качестве коэффициента изменения размера. Наконец, сгенерированный объект добавляется к полученному изображению сцены.

Модель Controlnet - Depth Version[27] используется для генерации сцен. На вход синтетической модели глубины поступает исходное изображение. Затем рассчитан-

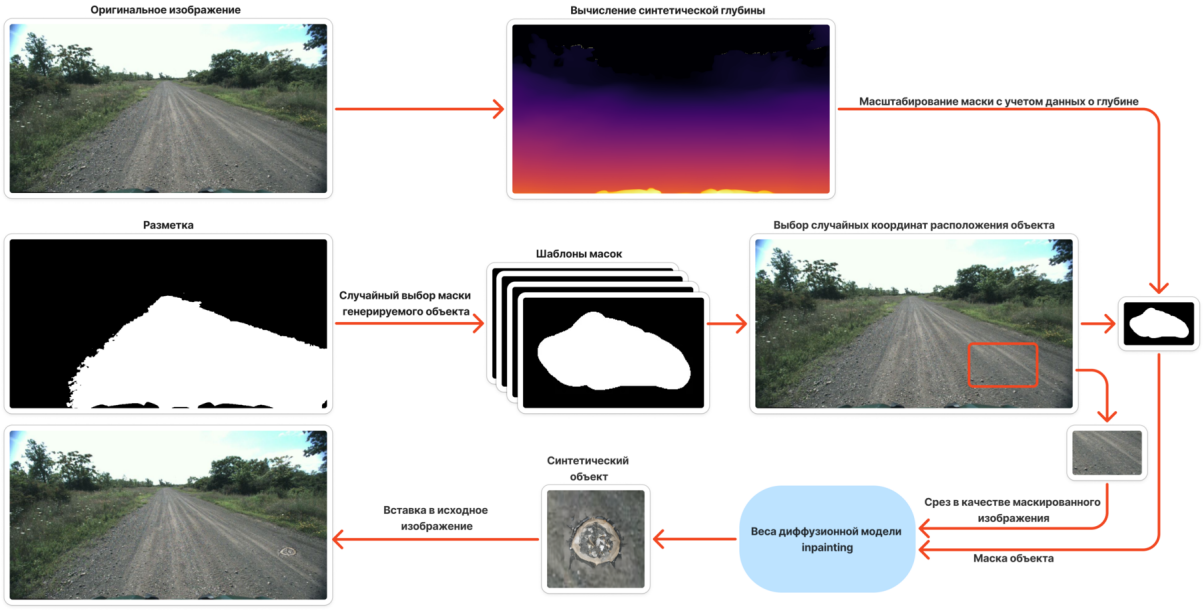

Рис. 12: Пайплайн генерации объектов.

ная синтетическая глубина подается на вход Controlnet и генерируется новое изображение сцены. Особенность заключается в том, что исходная разметка набора данных сохраняется, поэтому не требуется дополнительного времени.

Для понимания качества вычисления синтетической глубины, были использованы следующие метрики:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(z_i - \hat{z}_i)^2}, \quad \text{AbsRel} = \frac{1}{N}\sum_{i=1}^{N}\frac{|z_i - \hat{z}_i|}{z_i}$$

где z_i — истинное значение глубины, $\hat{z}_i$ — предсказанное значение.

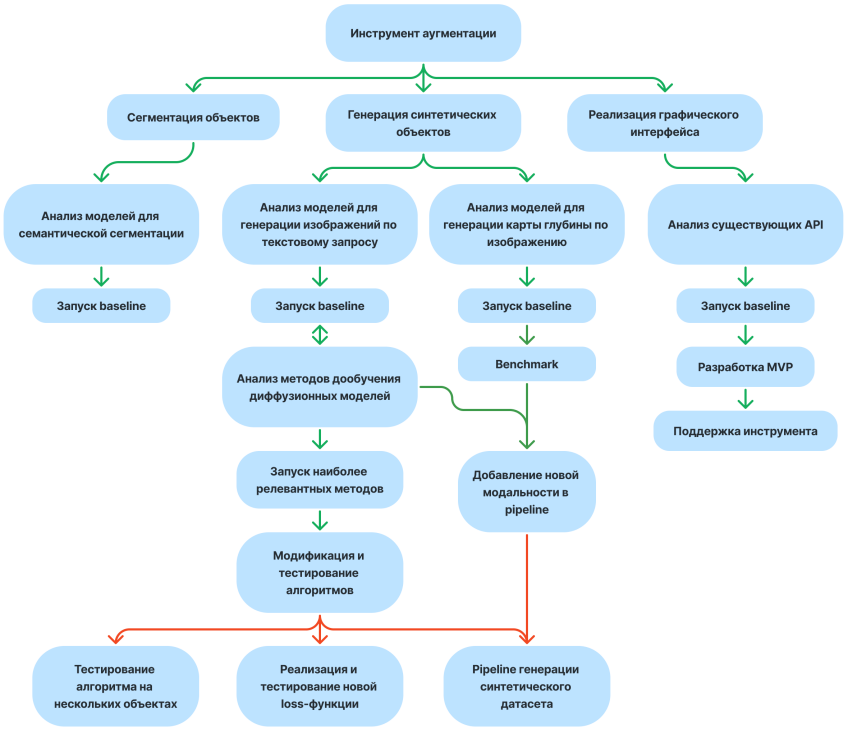

Рис. 13: Схема проведенных экспериментов.

4 Вычислительные эксперименты

4.1 Постановка экспериментов

В ходе работы эксперименты были поставлены как представлено на схеме 13. В качестве первых экспериментов были выбраны направления работы с моделями сегментации, генеративными моделями и реализация простого интерфейса для их использования.

Графический интерфейс. Также, для удобства тестирования моделей генерации изображений был реализован визуальный инструмент, который помогает проводить быстро эксперименты на малых данных и получать качественные результаты. Инструмент позволяет поделиться ссылкой на веб-приложение, используя встроенные в Gradio функции обмена без использования JavaScript, CSS или веб-хостинга. Пример работы приложения приведен на рисунках 14 и 15.

4.2 Эксперименты генерации объектов

Первые эксперименты были посвящены текстовой инверсии - вместо непосредственного дообучения весов моделей процесс оперирует текстовыми описаниями

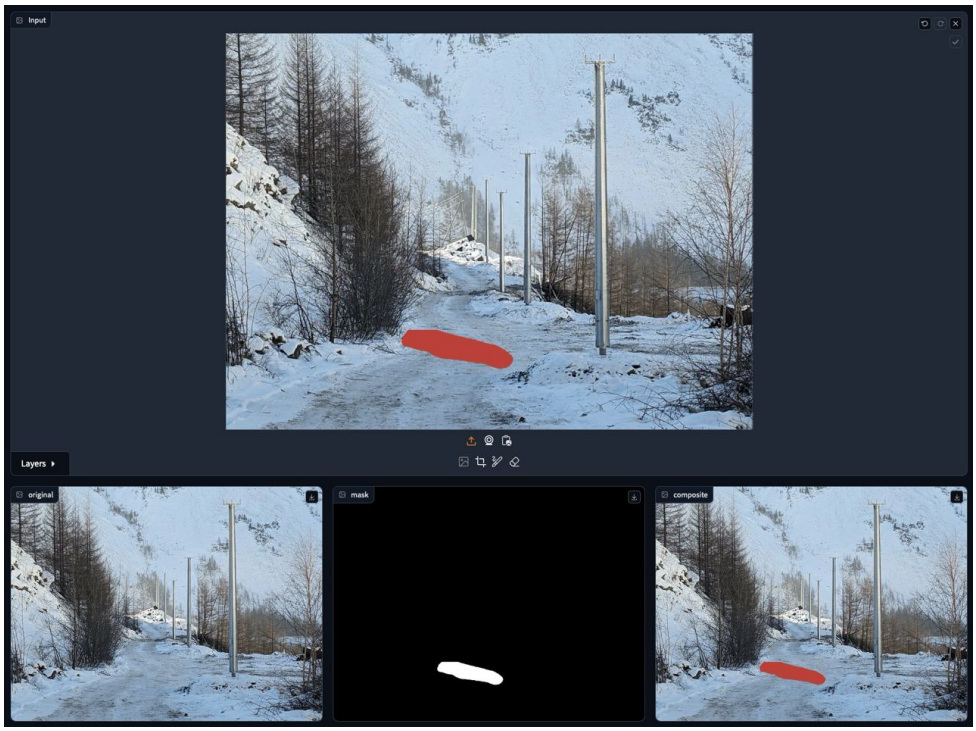

Рис. 14: Графический интерфейс. Пример получения маскированного изображения.

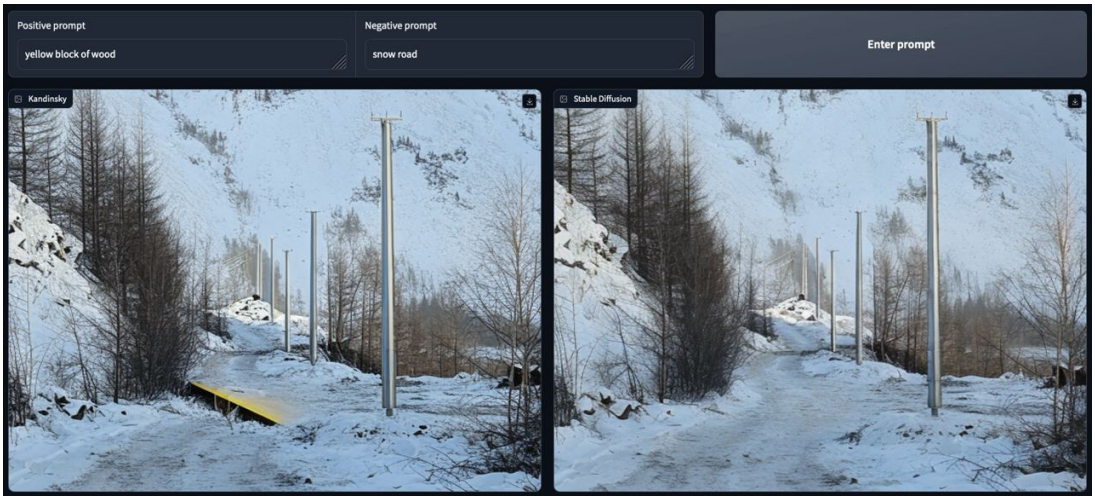

Рис. 15: Графический интерфейс. Пример аугментации изображения.

изображений для создания желаемых изменений или дополнений. Примеры сгенерированных изображений для разных текстовых запросов располагаются на рисунке 16.

Проблема заключается в том, что до недавнего времени текстовая инверсия не была использована для режима inpainting диффузионных моделей, несмотря на ее потенциальные преимущества. Хотя подход может быть полезным для inpainting, поскольку текстовое представление может содержать более сжатую и абстрактную

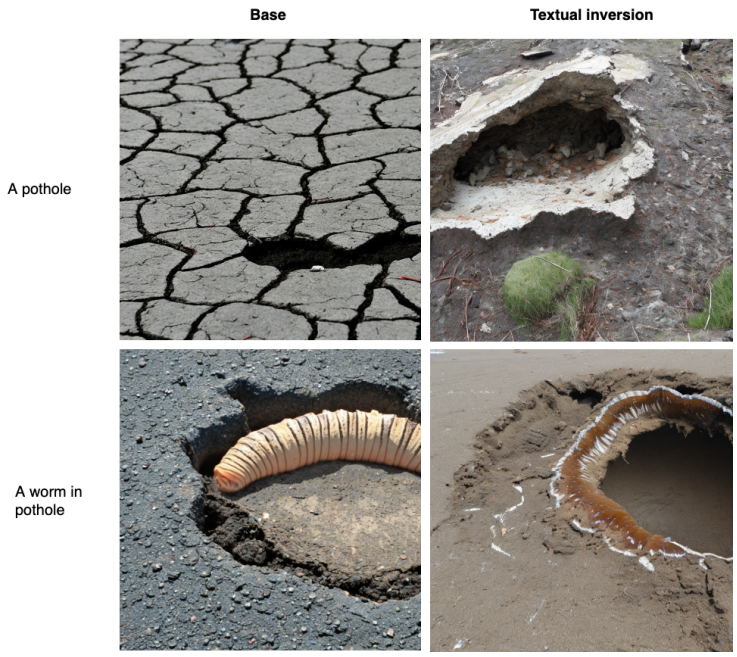

Рис. 16: Пример работы базовых весов с использованием текстовой инверсии на разных промптах. Дообучение осуществлялось на токен <pothole>.

информацию об изображении, что может помочь модели более точно заполнять пропущенные участки.

На схемах 7 и 9 представлены модифицированные алгоритмы текстовой инверсии и DreamBooth соответственно. На рисунке 17 представлены сводные результаты экспериментов дообучения базовых весов Stable Diffusion на данных алгоритмах. В первом столбце показано оригиналное изображение. Во втором столбце показана маскированная область. В третьем столбце показаны результаты генерации базовых весов. В четвертом столбце показаны результаты генерации базовых весов, дообученных с помощью текстовой инверсии. В пятом столбце показаны результаты генерации базовых весов, дообученных с помощью DreamBooth. Эксперименты проводились для двух классов: pothole (яма) - доменный класс; cat-avocado (мягкая игрушка) - очень узкий и специфичный класс, на котором можно провереть способность модели обобщать выученные знания.

Анализируя данный рисунок, можно прийти к следующим выводам:

- Все методы не имеют способности генерировать четкие и понятные человеку объекты;
- Исходные изображения практически не меняются, что говорит о том, что модели сложно понимать семантику изображения и что генерировать в данной

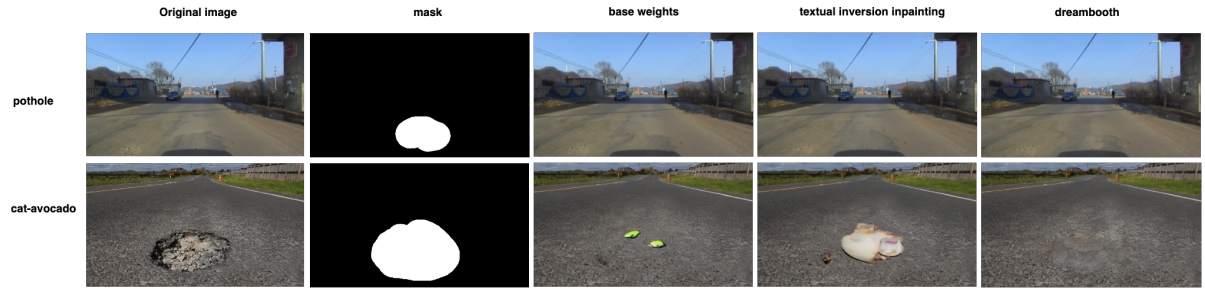

Рис. 17: Результаты экспериментов с использованием весов Stable Diffusion 2.

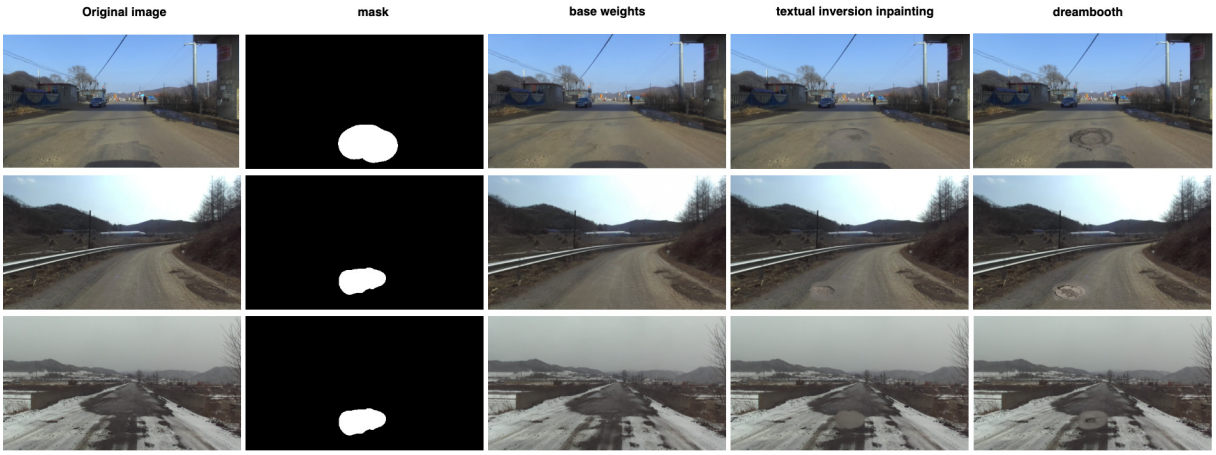

Рис. 18: Результаты экспериментов с использованием весов Stable Diffusion 2 и кропов изображений, включающих маску объекта.

области, поэтому она просто закрашивает выделенную область чем-то монотонным;

- В следстве 2 пунка абсолютно отсутствует вариативность.

Исходя из полученных результатов была выдвинута идея пайплайна аугментации изображений, представленная на схеме 12.

На рисунке 18 представлены результаты для тех же алгоритмов, но с использованием предложенного пайплайна.

На данном примере мы уже можем заметить, что:

- появились достаточно реалистичные объекты;
- на втором примере видно, что алгоритмы все так же чувствительны к изменению домена сцены (например, смена погодных условий);
- все так же отсутствует вариативность объектов.



Рис. 19: Инструмент фильтрации аннотаций датасетов.

Для решения проблемы низкой вариативности и генерации только одного класса было принято решение в дальнейших экспериментах использовать метод дообучения LoRA.

Сначала решалась проблема вариативности. Для этого необходимо было собрать достаточно большой датасет. Для упрощения задачи был выбран один класс - pothole (яма). Были выбраны три доменных датасета с платформы roboflow[1]: pothole dataset 1[2], pothole dataset 2[3], pothole dataset 3[4].

Однако, данные датасеты содержат много ненужных фотографий, который могут только ухудшить обучения модели. Для решения этой проблемы был реализован еще один графический интерфейс 19, с помощью которого можно было отфильтровать ненужные изображения и аннотации к ним.

После проведения процедуры фильтрации датасетов и их объединения был получен следующий единый датасет, со статистикой:

- Общее количество изображений: 833
- Общее количество аннотаций: 'pothole': 1800

[1] https://roboflow.com/

[2] https://universe.roboflow.com/dumani/yolda-cukur-inceleme

[3] https://universe.roboflow.com/smark-z7pr6/road-potholes-and-cracks-jmxtp/dataset/6

[4] https://universe.roboflow.com/jaya-rakesh-davuluri-fba0c/merge-project-lx9np/dataset/6

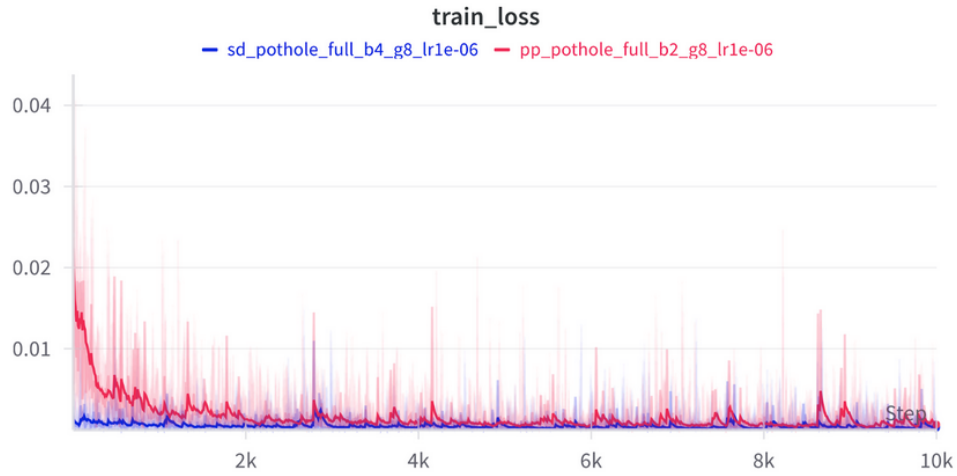

Рис. 20: График функции потерь для обучения Stable Diffusion и PowerPaint.

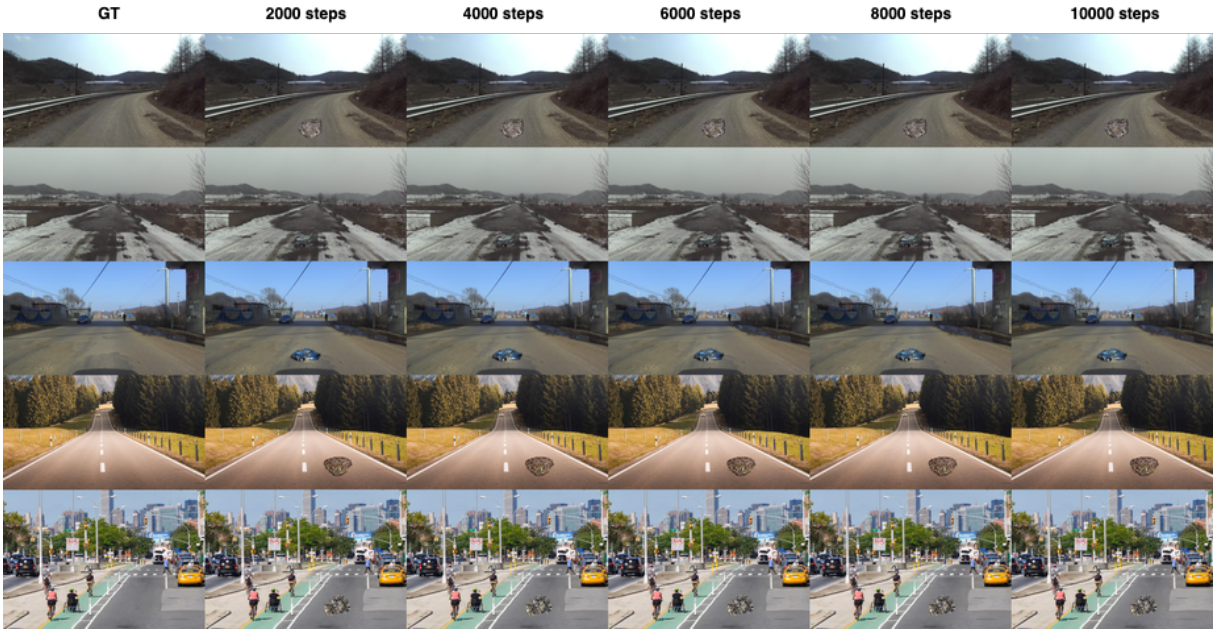

Рис. 21: Результаты экспериментов с использованием весов PowerPaint и кропов изображений, включающих маски объекта.

- Количество аннотированных изображений: 'pothole': 828

На рисунке 20 приведены графики обучения двух моделей на собранном датасете.

Первой была протестирована модель PowerPaint 21.

Наблюдая качественные результаты экспериментов, можно заметить, что объекты получаются слишком "утрированные достаточно яркие цвета, нереалистичные структуры объектов, которые режут глаз. Такие результаты являются неудовлетво-

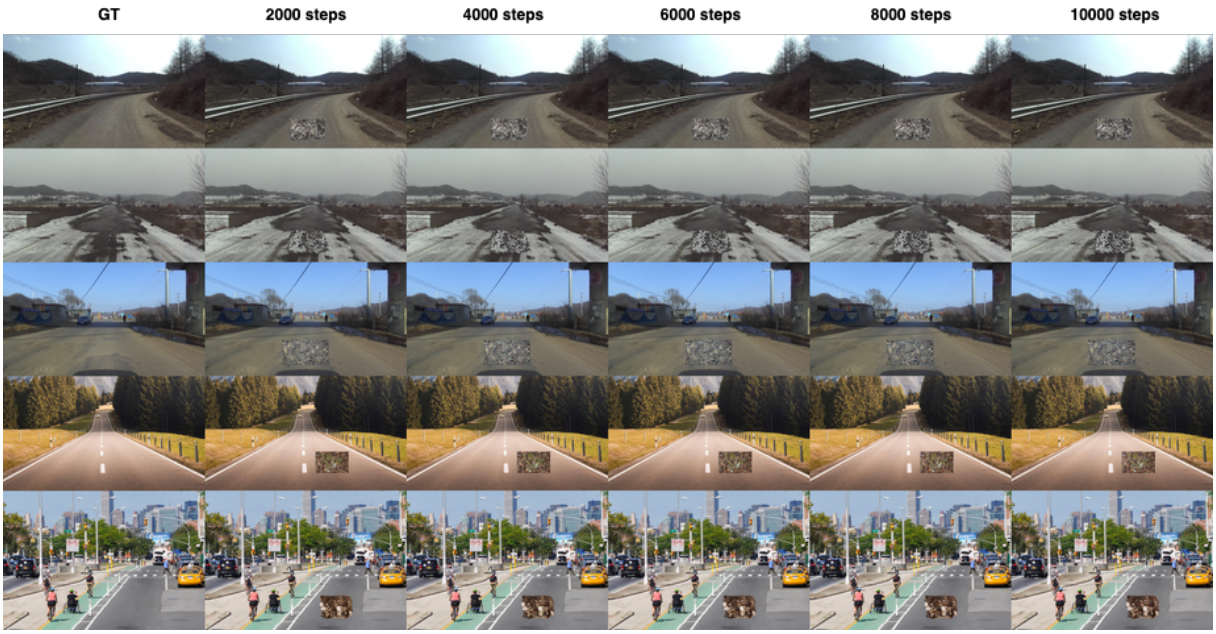

Рис. 22: Результаты экспериментов с использованием весов PowerPaint и кропов изображений, включающих bounding box объекта.

рительными для поставленной задачи.

Также, была выдвинута идея, что проблема может быть связана с видмо маски, подаваемой на вход - возможно, стоит использовать вместо маски объекта его бокс.

На рисунке 22 можно заметить, что данный подход только ухудшает ситуацию, поэтому было принято решение отказаться от использования модели PowePaint.

Затем была протестированна модель Stable Diffusion. На рисунке 23 представлены результаты ее обучения.

Анализируя результаты, делаем вывод, что:

- Объекты получаются достаточно реалистичные;
- Модель стала устойчива к изменению домена сцены;
- Появилась вариативность объектов на разных изображениях.

Делаем вывод, что задача решена успешно!

Теперь необходимо решить задачу введения нескольких классов объектов в генерацию. Также, необходимо получить достаточно разнообразный датасет. Для этого был выбран датасет TAOMR[32] 24.

На рисунке 25 приведен график обучения Stable Diffusion на выбранном датасете с помощью стандартной loss-функции - MSE.

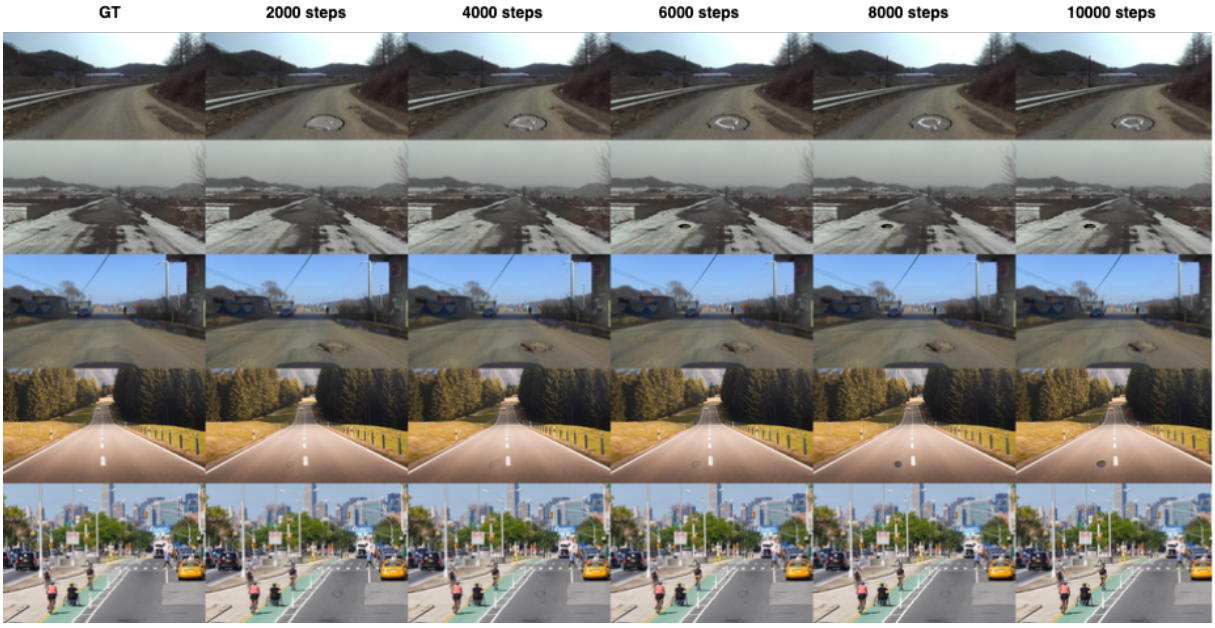

Рис. 23: Результаты экспериментов с использованием весов Stable Diffusion 2, обученные LoRA и кропов изображений, включающих маски объекта.

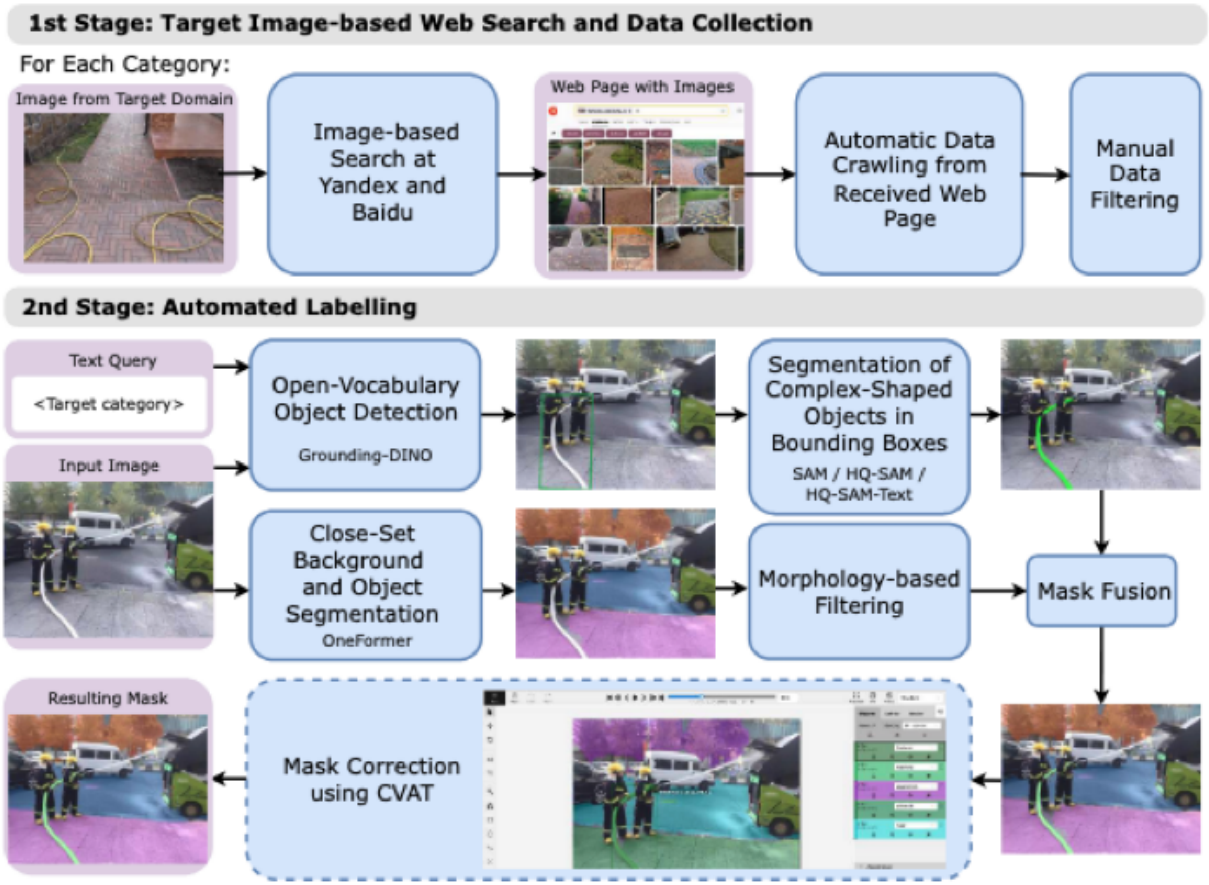

Рис. 24: Обзор датасета TAOMR[32].

Таблица 1: Количество автоматизированных меченых изображений в наборе данных TAOMR для каждого источника данных

Category	RR, Yandex and Baidu		NPO		InstSnowRoadDet		RoadObstacle21		Total	
	Images	Masks	Images	Masks	Images	Masks	Images	Masks	Images	Masks
Fire hose	272	387	0	0	0	0	0	0	272	387
Hose	269	328	0	0	0	0	0	0	269	328
Wire	220	231	0	0	0	0	0	0	220	231
Animal waste	239	1455	0	0	0	0	0	0	239	1455
Puddle	273	520	0	0	0	0	0	0	273	520
Road bump (pothole or speed bump)	2059	3174	2749	13964	84	216	310	2651	5202	20005
Sidewalk	1802	1802	2706	2706	0	0	243	243	4751	4751
Terrain	1753	1753	2839	2839	8	8	320	320	4920	4920
Vegetation	1711	1711	4981	4981	205	205	442	442	7339	7339
Road	2444	2444	5000	5000	240	240	442	442	8126	8126
Other obstacle negative	0	0	4571	10527	2	2	36	130	4609	10659
Other obstacle positive	0	0	3087	11788	194	743	437	4227	3718	16758
Total	3141	13805	5000	51805	240	1414	442	8455	8823	75479

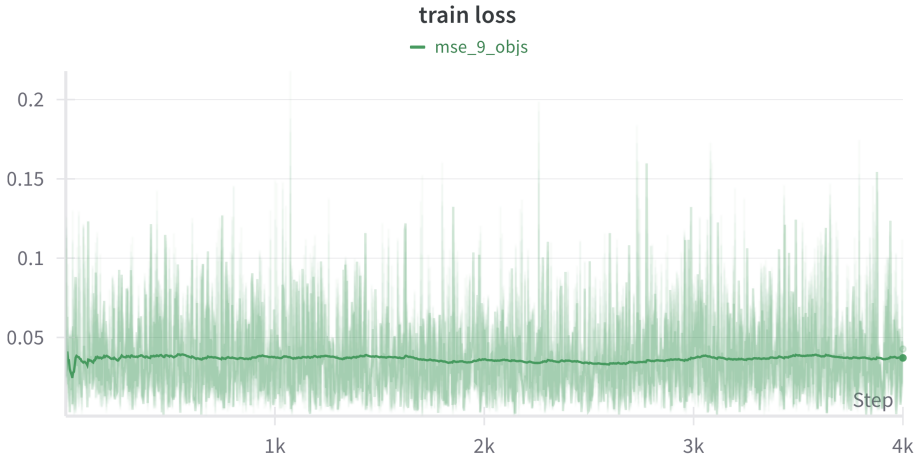

Рис. 25: График MSE функции потерь для обученися Stable Diffusion на датасете TAOMR[32].

Качественные результаты генерации объектов новых обученных весов приведены на рисунке 26.

Отметим, что на полученных результатах опять же заметны достаточно утри-



Рис. 26: Примеры генерации объектов на MSE функции потерь.

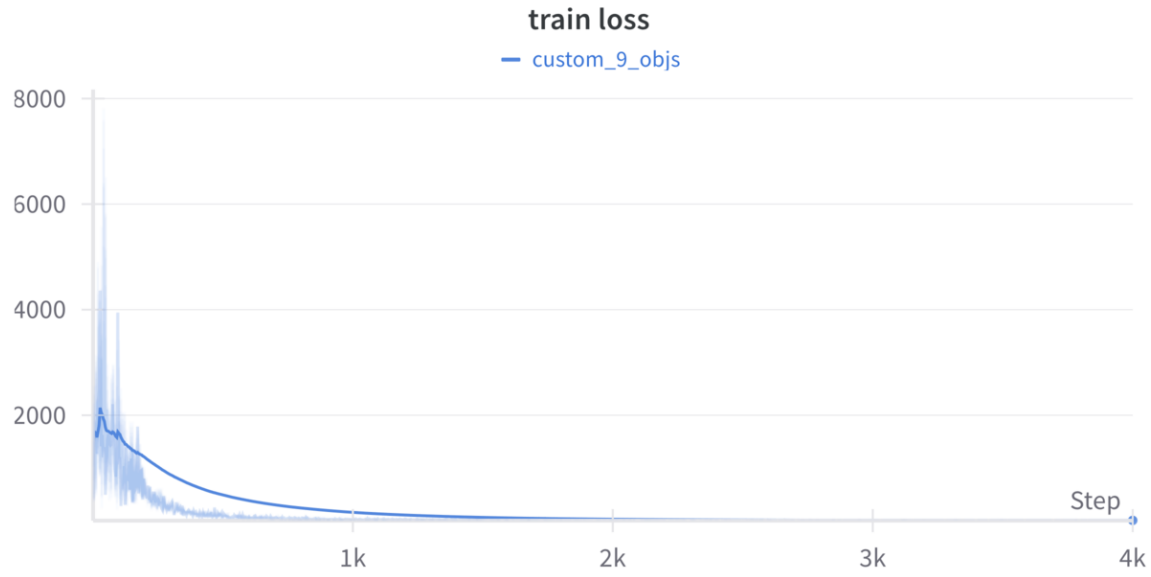

Рис. 27: График новой функции потерь для обучения Stable Diffusion на датасете TAOMR[32].

рованные результаты. Объекты перестали быть реалистичными.

Поэтому, были продолжены эксперименты уже с новой лосс функцией, процесс минимизации которой представлен на рисунке 27.

Качественные результаты генерации объектов новых обученных весов приведе-

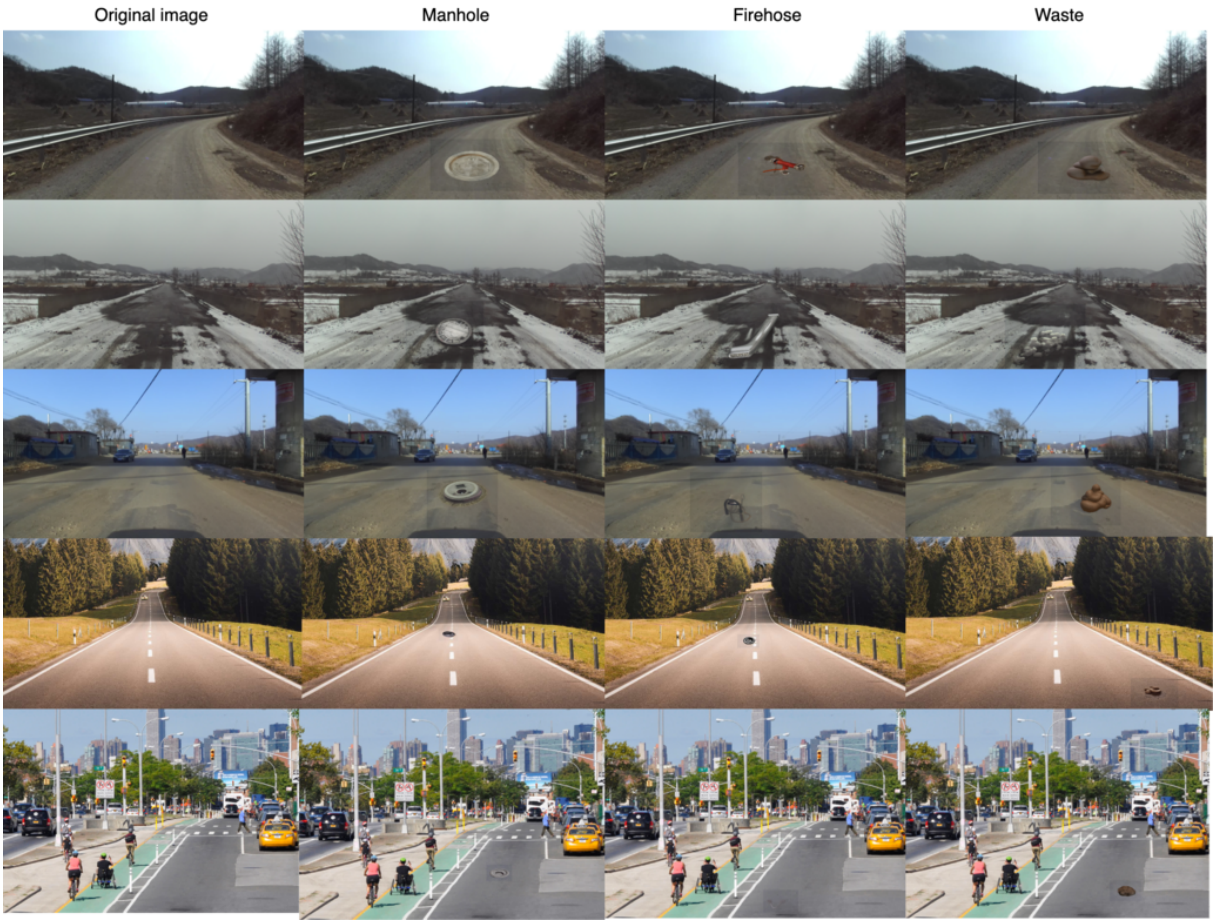

Рис. 28: Примеры генерации объектов на новой функции потерь.

ны на рисунке 28.

На полученных результатах видно, что объекты стали генерироваться лучше - видны их четкие и реалистичные структуры. Однако, появилась следующая проблема: модель начала затемнять немаскированную область. Модели при обучении стало выгоднее делать все пиксели изображения темнее, чтобы минимизировать второй член лосса.

Исходя из этого, было внесено изменение в предложеную функцию потерь (пункт 3)

Продолжая эксперименты с обучением, получаем новые графики минимизации функций потерб, которые представлены на рисунке 29.

Качественные результаты генерации объектов новых обученных весов приведены на рисунке 30.

В таблице 2 показаны метрики генерации изображений на тестовой выборке из набора данных TAOMR[32].Что улучшилось в RIC Loss: CLIP Score, SSIM, PSNR - все они немного увеличились. Это говорит о том, что сгенерированные изображения

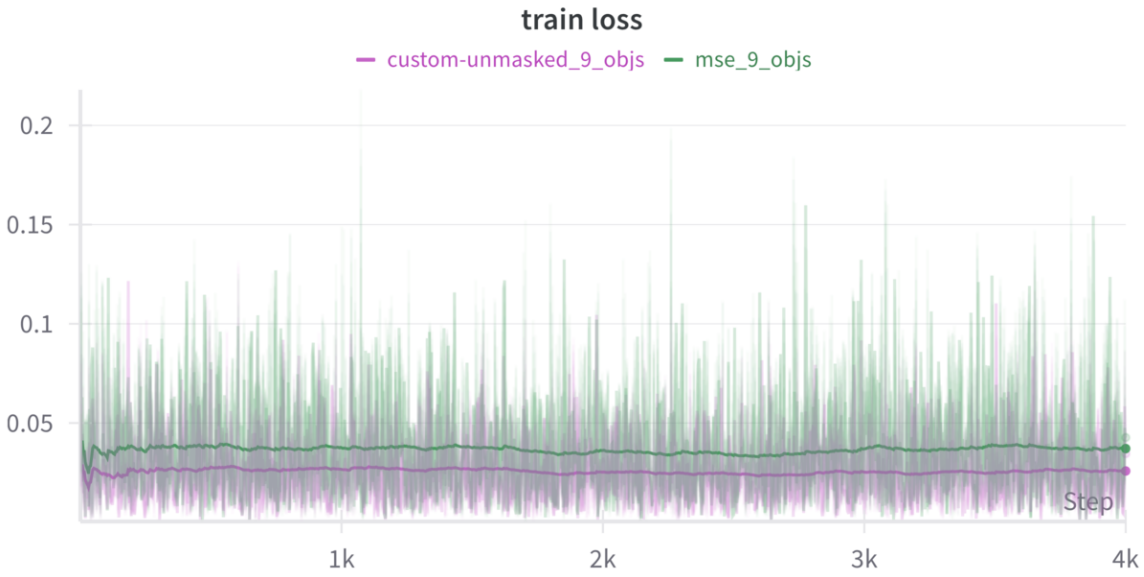

Рис. 29: График обновленной функции потерь для обученися Stable Diffusion на датасете TAOMR[32].



Рис. 30: Примеры генерации объектов на обновленной функции потерь.

Таблица 2: Metrics for the MSE and RIC Loss experiments

Loss	**CLIP Score** (↑)	**SSIM** (↑)	**PSNR** (↑)	**FID** (↓)	**LPIPS** (↓)
MSE	28.7500	0.1112	14.7453	113.6742	**0.5370**
IC LOSS	28.2969	0.0816	13.0155	**99.1669**	0.5677
RIC LOSS	**28.9219**	**0.1123**	**14.8449**	121.0460	0.5416

лучше соответствуют текстовым описаниям, имеют немного лучшую структуру и точность пикселей. Эти изменения незначительны (в пределах 1%), но все же положительны. Однако в RIC Loss FID и LPIPS обе метрики стали хуже. Это означает, что общее качество и визуальное сходство с реальными изображениями снизилось. Возможно, модель начала генерировать изображения, более точные локально (попиксельно), но хуже передающие глобальную семантику или разнообразие.

Метод RIC Loss в большей степени ориентирован на точное соответствие между текстом и локальной структурой изображения (выше CLIP, SSIM, PSNR). Но проигрывает в качестве и разнообразии (выше FID и LPIPS), т. е. результирующие изображения могут быть менее реалистичными или менее разнообразными. Метод MSE хуже по совпадению локального текста и точности пикселей, но лучше по качеству и визуальному сходству с реальными изображениями.

Общая интерпретация: метод RIC LOSS сильнее фокусируется на точном соответствии текста и локальной структуры изображения (выше CLIP, SSIM, PSNR). Но при этом теряет в качестве и разнообразии генерации (выше FID и LPIPS), то есть полученные изображения могут быть менее реалистичными или менее разнообразными. MSE хуже по локальному соответствию тексту и пиксельной точности, но лучше по качеству и визуальному сходству с реальными изображениями.

Проблема затемнения решена! Объекты остались так же реалистичными и вариативными, и при этом оставшаяся часть изображения не искажается.

Теперь с использованием полученных методов минимизации ошибки, можно тестировать модели сегментации изображений. Для экспериментов был выбран датасет YCOR.

В качестве модели сегментации использовались YOLOv8 medium и YOLOv11 medium. На таблице 3 приведены сводные результаты по обучению модели сегментации для класса pothole и manhole соответственно.

Видно, что предложенный метод аугментации датасетов дает прирост в метри-

Таблица 3: Metrics for the YOLOv8 and YOLO11 medium models

Model	Dataset	**Pothole**			**Manhole**		
		P (↑)	R (↑)	AP50 (↑)	P (↑)	R (↑)	AP50 (↑)
YOLOv8	TAOMR	0.600	0.302	0.404	<u>0.865</u>	0.176	<u>0.287</u>
YOLOv8	TAOMR + YCOR	0.576	0.334	0.450	0.587	0.176	0.210
YOLOv8	TAOMR + YCOR MSE	**0.606**	<u>0.391</u>	<u>0.450</u>	**1.000**	**0.215**	**0.288**
YOLOv8	TAOMR + YCOR RIC	<u>0.592</u>	**0.413**	**0.462**	0.738	<u>0.176</u>	0.209
YOLOv11	TAOMR	0.569	**0.365**	0.394	<u>0.745</u>	<u>0.173</u>	0.196
YOLOv11	TAOMR + YCOR	<u>0.709</u>	0.232	<u>0.444</u>	0.616	**0.176**	<u>0.211</u>
YOLOv11	TAOMR + YCOR MSE	0.560	0.242	0.401	0.621	0.118	0.207
YOLOv11	TAOMR + YCOR RIC	**0.668**	<u>0.333</u>	**0.453**	**1.000**	0.099	**0.242**

P: Precision, R: Recall

Название модели	**Среднее время (с, ↓)**	**RMSE** (↓)	**AbsRel** (↓)
Depth Anything	0.0380	9.1094	7.1462
Depth Anything V2	0.0384	8.9629	7.1368
MariGold	0.4283	10.4906	9.7801
DPT	0.0689	10.3837	6.9439

Таблица 4: Сравнительная табллица моделей вычисления синтетической глубины

ках. Однако, если сравнивать функции ошибки, то предложенная функция проигрывает слегка по метрикам только для положительного объекта.

4.3 Эксперименты генерации сцен

Для начала необходимо было понять какую модель вычисления синтетической глубины необходимо использовать в дальнейших экспериментах. Для этого были запущены жкспериментs на датасете YCOR, который содержит в себе информацию о реальной глубине, снятой со стерео камеры. На сравнительной таблице 4 видно, что лучше всего с данной задачей справляется модель Depth Anything V2. Она работает достаточно быстро и при этом имеет лучшие метрики по вычислению синтетической глубины.

По результатам бенчмарка моделей глубины были проведены результаты генерации синтетических сцен, которые представлены на рисунке 31.



Рис. 31: Результаты экспериментов с использованием весов Depth Anything V2 на датасете INSTSnowRoadDetection.

4.4 Генерация синтетического датасета

После всех проведенных экспериментов был сгенерирован полностью синтетический датасет на основе датасета INSTSnowRoadDetection.

Примеры создания полностью синтетического набора данных показаны на рисунке 32. Первый столбец содержит исходное изображение. Следующие столбцы содержат изображения для $guidencescale = [0.7, 1, 2, 3]$ соответственно. Для каждого исходного изображения из набора данных генерируется одно синтетическое изображение с одним объектом (выбоиной). Всего получено 842 изображений из выборки *train*, 38 изображений в *validation* и 24 изображений в *test*.

4.5 Стек технологий

При проведении экспериментов были использованы следующие основные библиотеки и фреймворки:

- diffusers - фреймворк для генерации изображений;
- transformers - библиотека для работы с трансформенными структурами;
- pycocotools - инструменты для работы с COCO-датасетом;

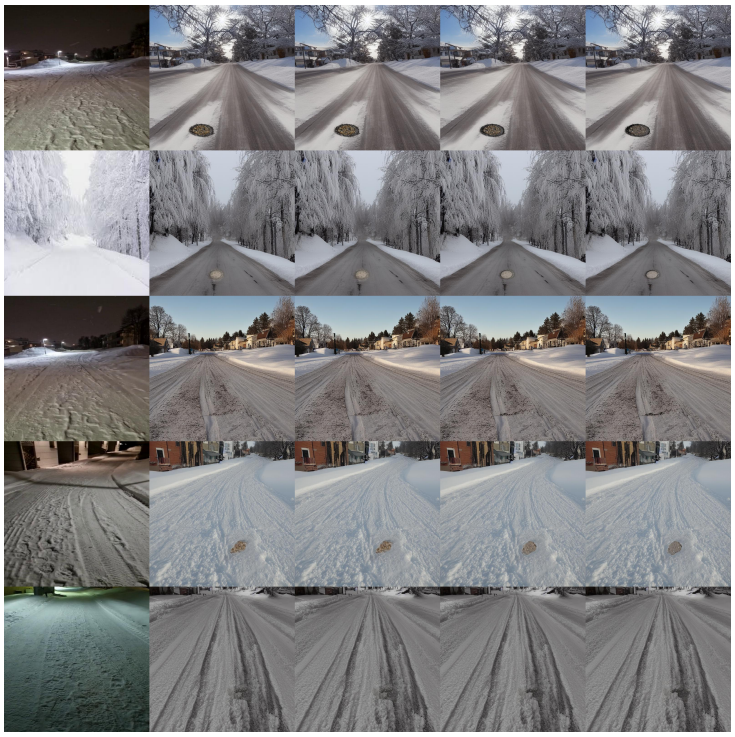

Рис. 32: Результаты генерации синтетического датасета с использованием в Depth Anything V2 и Stable Diffusion 1.5

- ultralytics - библиотека для работы с YOLO (обнаружение объектов'
- Gradio - это пакет Python с открытым исходным кодом, который позволяет быстро создать демонстрацию или веб-приложение для вашей модели машинного обучения, API или любой произвольной функции Python;
- PyQT - фреймворк для создания графических интерфейсов на Python;

Все эксперименты проводились на двух NVIDIA Tesla V100 32GB.

4.6 Выводы

После проведения большого обзора литературы, изучения теоретических материалов и постановки всех экспериментов, можно сделать следующие выводы:

- Предложенная loss-функция в отличии от базовой MSE дает меньшую ошибку при обучении и более качественные объекты при генерации на валидации
- Предложенный метод аугментации увеличивает метрики сегментации
- Генерация сцен в связи с генерацией объектов позволяет сгенерировать множество новых данных в нужном домене

5 Заключение

Были осуществлены следующие этапы:

- Реализация новой loss-функции для обучения диффузионных моделей
- Реализация пайплайна генерации полностью синтетического датасета

В дальнейших исследованиях планируется осуществить следующие этапы работы:

- Улучшение метрик сегментации на аугментированном датасете
- Тестирование моделей сегментации на полностью синтетическом датасете
- Автоматизация пайплайна генерации синтетического датасета

По итогам работы были опубликованы следующие статьи:

- Izmestyeva U. et al, (2025), 'DataEdit: Dataset editor using diffusion inpainting' - планируется подача на конференцию 'Нейроинформатика'.
- Vetoshkin L. et al. How to Segment Wires, Hoses, and Other Obstructive Objects in the Traversable Area of a Mobile Robot //Hoses, and Other Obstructive Objects in the Traversable Area of a Mobile Robot.
- Staroverov A. et al. Fine-tuning multimodal transformer models for generating actions in virtual and real environments //Ieee Access. – 2023. – Т. 11. – С. 130548-130559.

Список литературы

1. *Поспелов Д. А.* Ситуационное управление: теория и практика. — М. : Наука, 1986. — С. 288.
2. *Гаазе-Рапопорт М. Г.*, *Поспелов Д. А.* От амебы до робота. Модели поведения. — М. : Наука, 1987. — С. 288.
3. *Поспелов Д. А.* Моделирование рассуждений. Опыт анализа мыслительных актов. — М. : Радио и связь, 1989. — С. 184.
4. *Осипов Г. С.* Приобретение знаний интеллектуальными системами. — М. : Наука. Физматлит, 1997. — С. 112.
5. *Осипов Г. С.*, *Поспелов Д. А.* Прикладная семиотика // Новости искусственного интеллекта. — 1999. — № 1. — С. 9—35.
6. Microsoft coco: Common objects in context / T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick // Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. — Springer. 2014. — С. 740—755.
7. *Ronneberger O.*, *Fischer P.*, *Brox T.* U-net: Convolutional networks for biomedical image segmentation // Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. — Springer. 2015. — С. 234—241.
8. *Осипов Г. С.* Методы искусственного интеллекта. — 2-е изд. — М. : ФИЗМАТЛИТ, 2015. — С. 297.
9. *Gupta A.*, *Dollar P.*, *Girshick R.* Lvis: A dataset for large vocabulary instance segmentation // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. — 2019. — С. 5356—5364.
10. End-to-end object detection with transformers / N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko // European conference on computer vision. — Springer. 2020. — С. 213—229.
11. Learning transferable visual models from natural language supervision / A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark [и др.] // International conference on machine learning. — PMLR. 2021. — С. 8748—8763.

12. Mdetr-modulated detection for end-to-end multi-modal understanding / A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, N. Carion // Proceedings of the IEEE/CVF International Conference on Computer Vision. — 2021. — С. 1780—1790.

13. OpenCLIP / G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, L. Schmidt. — Вер. 0.1. — 07.2021. — DOI: 10.5281/zenodo.5143773. — URL: https://doi.org/10.5281/zenodo.5143773 ; If you use this software, please cite it as below.

14. *Ranftl R.*, *Bochkovskiy A.*, *Koltun V.* Vision Transformers for Dense Prediction // ArXiv preprint. — 2021.

15. An image is worth one word: Personalizing text-to-image generation using textual inversion / R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, D. Cohen-Or // arXiv preprint arXiv:2208.01618. — 2022.

16. Hierarchical text-conditional image generation with clip latents / A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, M. Chen // arXiv preprint arXiv:2204.06125. — 2022. — Т. 1, № 2. — С. 3.

17. High-resolution image synthesis with latent diffusion models / R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. — 2022. — С. 10684—10695.

18. Lora: Low-rank adaptation of large language models. / E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen [и др.] // ICLR. — 2022. — Т. 1, № 2. — С. 3.

19. Photorealistic text-to-image diffusion models with deep language understanding / C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans [и др.] // Advances in neural information processing systems. — 2022. — Т. 35. — С. 36479—36494.

20. Simple open-vocabulary object detection / M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen [и др.] // European Conference on Computer Vision. — Springer. 2022. — С. 728—755.

21. A Simple Framework for Open-Vocabulary Segmentation and Detection / H. Zhang, F. Li, X. Zou, S. Liu, C. Li, J. Gao, J. Yang, L. Zhang // arXiv preprint arXiv:2303.08131. — 2023.

22. *DEBIGP*. INSTSnowRoadDetection Dataset : Open Source Dataset. — 12.2023. — URL: `https://universe.roboflow.com/debigp/instsnowroaddetection` ; visited on 2025-04-07. `https://universe.roboflow.com/debigp/instsnowroaddetection`.

23. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation / N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, K. Aberman // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2023. — С. 22500—22510.

24. Grounding dino: Marrying dino with grounded pre-training for open-set object detection / S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu [и др.] // arXiv preprint arXiv:2303.05499. — 2023.

25. Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion / A. Razzhigaev, A. Shakhmatov, A. Maltseva, V. Arkhipkin, I. Pavlov, I. Ryabov, A. Kuts, A. Panchenko, A. Kuznetsov, D. Dimitrov // arXiv preprint arXiv:2310.03502. — 2023.

26. Multi-concept customization of text-to-image diffusion / N. Kumari, B. Zhang, R. Zhang, E. Shechtman, J.-Y. Zhu // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2023. — С. 1931—1941.

27. *Zhang L., Agrawala M.* Adding Conditional Control to Text-to-Image Diffusion Models. — 2023. — arXiv: `2302.05543 [cs.CV]`.

28. Depth anything v2 / L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, H. Zhao // Advances in Neural Information Processing Systems. — 2024. — Т. 37. — С. 21875—21911.

29. Depth anything: Unleashing the power of large-scale unlabeled data / L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, H. Zhao // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2024. — С. 10371—10381.

30. Depth Pro: Sharp Monocular Metric Depth in Less Than a Second / A. Bochkovskii, A. Delaunoy, H. Germain, M. Santos, Y. Zhou, S. R. Richter, V. Koltun // arXiv. — 2024.

31. Marigold: Affordable Adaptation of Diffusion-Based Image Generators for Image Analysis / B. Ke, K. Qu, T. Wang, N. Metzger, S. Huang, B. Li, A. Obukhov, K. Schindler. — 2025. — arXiv: 2505.09358 [cs.CV].

32. How to Segment Wires, Hoses, and Other Obstructive Objects in the Traversable Area of a Mobile Robot / L. Vetoshkin, O. Matykina, M. Kichik, U. Izmesteva, D. Yudin // Hoses, and Other Obstructive Objects in the Traversable Area of a Mobile Robot. —.