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

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

УДК 004.93

**Отчет об исследовательском проекте на тему:**

# Приложение генеративных моделей для решения дискриминативных задач

**Выполнил студент:**

группы #БПМИ211, 3 курса Матосян Александр Акобович

**Принял руководитель проекта:**

Мещанинов Вячеслав Павлович
Научный сотрудник
Факультет компьютерных наук НИУ ВШЭ

**Соруководитель:**

Аланов Айбек
Научный сотрудник
Факультет компьютерных наук НИУ ВШЭ

**Консультант:**

Находнов Максим

Москва 2024

# Содержание

# Аннотация

Задача семантической сегментации изображений является базовой задачей компьютерного зрения, сутью которой является отнесение каждого пикселя изображения к какому-нибудь классу. Данная курсовая работа нацелена на исследование методов решения этой задачи с помощью генеративных диффузионных моделей. Для начала были изучены сами диффузионные модели, затем был исследован существующий метод применения таких моделей для решения задачи семантической сегментации изображений. Основной идеей метода является использование диффузионной модели, как способа получения новых представлений для пикселей исходного изображения. Также, были выявлены слабые места исследуемого метода, на основе которых были выдвинуты и проверены гипотезы для его улучшения. Результатом курсовой работы является модификация, основанная на аугментациях изображений.

# Ключевые слова

Глубинное обучение, компьютерное зрение, генеративные модели, диффузионные модели

# 1 Введение

## 1.1 Описание предметной области

Генеративные модели являются одним из самых перспективных направлений в области глубинного обучения и находят применение в различных задачах, в основном связанных с генерацией данных различного типа. Особый интерес представляют диффузионные генеративные модели, в основе которых лежит идея диффузии из физики. Они моделируют распределение данных и позволяют генерировать образцы высокого качества из этого распределения. Однако, диффузионные модели можно также применять для решения негенеративных задач, например для семантической сегментации изображений.

Одной из классических задач компьютерного зрения является семантическая сегментация - процесс разделения изображения на отдельные сегменты или регионы, каждый из которых относится к определенному классу. Эта задача играет важную роль в области компьютерного зрения и обработки изображений, позволяя системам компьютерного зрения более полно и точно понимать их содержание и контекст.

## 1.2 Постановка задачи

Несмотря на то, что задача сегментации хорошо решается с помощью сверточных нейросетей и трансформеров, можно взглянуть на неё под другим углом и попытаться решить её используя генеративные диффузионные модели. В рамках данной курсовой работы предлагается исследовать метод сегментации эксплуатирующий диффузионные модели.

Для начала требовалось изучить основные концепции в диффузионных моделях и теорию стоящую за ними, а также базовые статьи, исследующие их.

Далее нужно было рассмотреть существующий SOTA подход применения диффузионных моделей для решения задачи семантической сегментации. Были выделены слабые места подхода. В результате глубокого исследования метода, была описана его модификация, полученная путем проверки гипотез по улучшению. Всё исследование должно подкреплено эмпирическими данными и сравнительным анализом.

# 2 Обзор литературы

## 2.1 Deep Unsupervised Learning using Nonequilibrium Thermodynamics(вспомогательная статья)

Авторы статьи [5] представляют новый подход к машинному обучению, основанный на принципах из статистической физики для создания эффективной и гибкой генеративной модели данных. В основе метода лежит итеративный процесс, в ходе которого простое распределение данных систематически трансформируется в более сложное через процесс прямой диффузии, а затем модель обучается выполнению обратного диффузионного процесса. Этот обратный процесс способствует восстановлению структуры данных. Данный подход эффективно аппроксимирует неизвестное распредленение данных и позволяет генерировать его образцы. Модель способна обрабатывать сложные наборы данных, а также эффективно вычислять условные и апостериорные вероятности. В статье приводятся результаты экспериментов с различными данными, демонстрирующие преимущества этого метода в моделировании сложных распределений данных.

## 2.2 Denoising Diffusion Probabilistic Models (базовая статья)

### 2.2.1 Основная идея

Данная статья [2] представляет собой важный вклад в область генеративного моделирования изображений с использованием диффузионных методов. Авторы статьи предлагают новый подход к генерации изображений, основанный на диффузионном процессе, который позволяет получать высококачественные изображения с помощью последовательного расшумления шумных изображений.

Для решения задачи генерации предлагается аппроксимировать распределение данных(точнее его плотность), имея доступ к которому можно будет семплировать новые объекты. Одним из способов семплирования является динамика Ланжевена:

$$x^{(n)} = x^{(n-1)} + \varkappa \left( \frac{1}{2} \frac{\partial}{\partial x} \log p(x) + \frac{\varepsilon}{\sqrt{\varkappa}} \right), \varepsilon \sim \mathcal{N}(\varepsilon|0, I)$$

Авторы предлагают другой метод семплирования и описывают модель диффузии через forward process и reverse process. При forward process объект из данных постепенно за-

шумляется. Математически это описывается так:

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\varepsilon, \varepsilon \sim \mathcal{N}(\varepsilon|0, I)$$

$\beta_t$ регулирует шум, впрыскиваемый в объект, а также скорость разрушения. При $t \to \infty$ распределение $x_t \sim \mathcal{N}(x_t|0, I)$.

Отметим, что в forward process нет обучаемых параметров, и известны распределения зашумленных объектов для всех моментов времени:

$$q\left(x_t \mid x_{t-1}\right) := \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right)$$

$$q(x_t|x_0) = \mathcal{N}(x_t|; \sqrt{\overline{\alpha_t}}x_0, (1-\overline{\alpha_t})I), \overline{\alpha_t} = \prod_{s=1}^{t} \alpha_s, \alpha_t = 1 - \beta_t$$

Сутью же обратного процесса является получение образца из искомого распределения путём постепенного расшумления объекта $x^{(T)} \sim \mathcal{N}(x^{(T)}|0, I)$ (то есть белого шума). Все обучаемые параметры находятся в этом процессе, а именно, модель должна приближать распределения, описывающие переход от шумного объекта к менее шумному, для каждого момента времени t.

### 2.2.2 Функция потерь и параметризация модели

При обучении оптимизируется NLLLoss(Negative Log Likelihood Loss). Авторы выводят верхнюю оценку на функцию потерь и минимизируют её:

$$\mathbb{E}\left[-\log p_\theta\left(\mathbf{x}_0\right)\right] \leq \mathbb{E}_q\left[D_{KL}(q(x_T|x_0)||p(x_T)) + \sum_{t>1} D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) - \log p_\theta(x_0|x_1)\right]$$

В верхней оценке можно заметить, что нужно минимизировать лишь сумму KL-дивергенций прямых и обратных переходов.

Авторы предлагают две параметризации: среднее гауссианы, отвечающей за переход в обратном процессе, либо впрыскиваемый шум. Второй способ параметризации показал себя лучше. В таком случае, модель предсказывает шум, который был добавлен к объекту на t-ом шаге. Авторы отмечают, что модель в таком виде более численно устойчива. Для такой параметризации функция потерь принимает довольно простой вид:

$$L_{simple}(\theta) := \mathbb{E}_{t,x_0,\varepsilon}\left[||\varepsilon - \varepsilon_\theta(\sqrt{\overline{\alpha_t}}x_0 + \sqrt{1-\overline{\alpha_t}}\varepsilon, t)||^2\right]$$

Такая параметризация позволяет вывести простое выражение для поиска $x_{t-1}$ при наличии $x_t$:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}} \varepsilon_\theta(x_t, t) \right) + \sigma_t z, z \sim \mathcal{N}(0, I)$$

За аппроксимацию впрыскиваемого шума отвечает нейронная сеть, которая принимает на вход момент времени t и объект $x_t$ и предсказывает шум $\varepsilon_\theta(x_t, t)$. В своей реализации авторы используют нейронную сеть с архитектурой U-Net.

Алгоритмы обучения и генерации при параметризации через шум:

**Algorithm 1** Training

**repeat**
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \text{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
    $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
**until** converged

**Algorithm 2** Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
**for** $t = T, \dots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, )$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
**end for**
**return** $\mathbf{x}_0$

### 2.2.3 Количественные результаты

В статье авторы демонстрируют результаты своих экспериментов на датасете CIFAR10 и приводят метрики Inception score и FID score(Fréchet inception distance). В задаче безусловной генерации, данный метод показывает лучший FID score.

## 2.3 High-Resolution Image Synthesis with Latent Diffusion Models(вспомогательная статья)

Метод, описанный в статье [4], базируется на использовании диффузионных моделей в латентном пространстве, что позволяет снизить вычислительные затраты и одновременно сохранить высокое качество в процессе генерации изображений. Основные новшества статьи:

- В архитектуре метода присутствуют энкодер и декодер блоки, которые отвечают за перевод данных в латентное пространство и из латентного пространства соответственно. Авторы используют такую концепцию для уменьшения сложности вычислений.
- Вместо работы с пикселями напрямую, диффузионная модель обучается восстанавливать исходные изображения из зашумлённых данных в латентном пространстве. Это значительно снижает вычислительные затраты и ускоряет процесс генерации изображений.
- В архитектуру модели были добавлены слои внимания (cross-attention layers). Это было сделано для того, чтобы генерация могла быть обусловлена различными условиями, например текстом.

Таким образом, описанный авторами метод позволяет решать задачу условной генерации изображений с меньшими вычислительными затратами.

## 2.4 Label-Efficient Semantic Segmentation with Diffusion Models (основная статья)

### 2.4.1 Основная идея

Модели типа DDPM(Denoising Diffusion Probabilistic Models) в последнее время получили значительное внимание исследователей, поскольку они превосходят альтернативные подходы, такие как генеративно-состязательные сети (GAN), и, в настоящее время, являются state-of-the-art решением в области генерации. Такие модели имеют множество приложений, включая восстановление изображений с пропущенными сегментами, увеличение разрешения изображений и семантическое редактирование. Авторы данной статьи [1] демонстрируют, что DDPM также могут быть использованы для решения задачи семантической сегментации изображений, особенно в ситуации, когда размеченных данных недостаточно.

Авторы описывают простой метод сегментации. Для решения данной задачи предлагается использовать промежуточные активации предобученной сети, которая параметризует

reverse process в DDPM. На их основе для каждого пикселя исходного изображения формируются новые представления, которые дальше используются для определения класса пикселя.

### 2.4.2 Архитектура

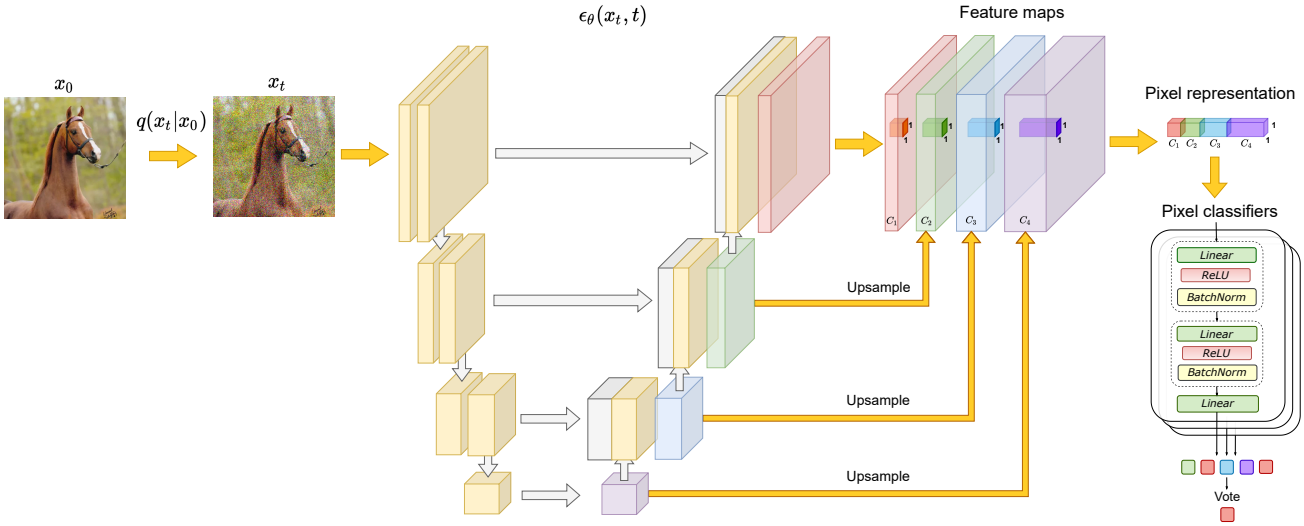



Рис. 2.1: **Обзор предложенного метода.** (1) $x_0 \to x_t$ путем добавления шума в соответствии с $q(x_t|x_0)$. (2) Извлечение карт признаков из нейронной сети, параметризующей шум $\epsilon_\theta(x_t, t)$. (3) Формирование новых представлений пикселей путем конкатенации и увеличения размерности карт признаков. (4) Использование новых представлений пикселей для обучения ансамбля MLP сетей, определяющих к какому классу относится конкретный пиксель.

В DDPM за параметризацию модели отвечает нейронная сеть с архитектурой U-Net. Авторы используют уже предобученную такую сеть. Сам метод сегментации можно описать в несколько пунктов:

- На вход U-Net подается зашумленная версия сегментируемого изображения.
- Берутся некоторые из полученных промежуточных слоев, и, с помощью повышения размерности и конкатенации, формируются представления для каждого пикселя сегментируемого изображения.
- Полученные для пикселей представления пропускаются через ансамбль MLP классификаторов для определения пикселя к конкретному классу.

Стоит отметить, что авторы обучали только ансамбль классификаторов. За обучающую выборку бралась маленькая часть датасета, на котором был обучен backbone, и размечалась для задачи сегментации. Одним из главных достоинств предложенного метода как раз таки является то, что классификаторы можно обучить на очень маленьком количестве размеченных изображений. Для MLP классификаторов авторы берут не глубокую архитектуру с двумя скрытыми слоями, каждый из которых сопровождается слоем нелинейности ReLU и batch нормализацией.

### 2.4.3 Количественные результаты

Для определения качества сегментации использовалась метрика IoU(Intersection of Union).

Для экспериментов использовались следующие наборы данных: Bedroom-28, FFHQ-34, Cat-15, Horse-21, CelebA-19, ADE-Bedroom-30. Также, эксперименты ставились на синтетичсеких данных, которые сгенерировали с помощью DDPM и GAN.

Экспериментальным путём, авторы пришли к тому, что представления получаемые на более поздних шагах reverse process-а DDPM хранят в себе больше семантической информации об исходном изображении. Также, стоит отметить, что более широкие слои U-Net сети дают более информативные представления, чем узкие слои.

| Method | Bedroom-28 | FFHQ-34 | Cat-15 | Horse-21 | CelebA-19* | ADE Bedroom-30* |
|---|---|---|---|---|---|---|
| ALAE | 20.0 ± 1.0 | 48.1 ± 1.3 | — | — | 49.7 ± 0.7 | 15.0 ± 0.5 |
| VDVAE | — | 57.3 ± 1.1 | — | — | 54.1 ± 1.0 | — |
| GAN Inversion | 13.9 ± 0.6 | 51.7 ± 0.8 | 21.4 ± 1.7 | 17.7 ± 0.4 | 51.5 ± 2.3 | 11.1 ± 0.2 |
| GAN Encoder | 22.4 ± 1.6 | 53.9 ± 1.3 | 32.0 ± 1.8 | 26.7 ± 0.7 | 53.9 ± 0.8 | 15.7 ± 0.3 |
| SwAV | 42.4 ± 1.7 | 56.9 ± 1.3 | 45.1 ± 2.1 | 54.0 ± 0.9 | 52.4 ± 1.3 | 30.6 ± 1.6 |
| MAE | 45.0 ± 2.0 | **58.8 ± 1.1** | **52.4 ± 2.3** | 63.4 ± 1.4 | 57.8 ± 0.4 | 31.7 ± 1.8 |
| DatasetGAN | 31.3 ± 2.3 | 57.0 ± 1.1 | 36.5 ± 2.3 | 45.4 ± 1.4 | — | — |
| DatasetDDPM (Ours) | 47.9 ± 2.9 | 56.0 ± 0.9 | 47.6 ± 1.5 | 60.8 ± 1.0 | — | — |
| **DDPM (Ours)** | **49.4 ± 1.9** | **59.1 ± 1.4** | **53.7 ± 3.3** | **65.0 ± 0.8** | **59.9 ± 1.0** | **34.6 ± 1.7** |

Таблица 2.1: Сравнение методов сегмнетации по метрике IoU.

По таблице 2.1 с метриками можно заметить, что предложенный авторами метод показывает наилучшие результаты на всех наборах данных

### 2.5 Semantic Segmentation with Generative Models: Semi-Supervised Learning and Strong Out-of-Domain Generalization (вспомогательная статья)

Метод, описанный в статье [3], использует другой вид генеративных моделей для решения задачи семантической сегментации. Выбор авторов пал на генеративные состязательные сети (GAN), которые в данном случае используются для моделирования совместного распределения изображений и меток, что позволяет генерировать как изображения, так и их маски сегментации. Архитектура модели построена на базе StyleGAN2, дополненного модулем генерации меток. Модель обучается на большом наборе неразмеченных изображений и небольшом количестве размеченных данных. При обучении, для сгенерированных изображений используется Reconstruction Loss, а для масок Supervised Loss.

На этапе предсказания маски получаются следующим образом: для сегментируемого изображения получается эмбеддинг в совместном латентном пространстве с помощью encoder-сети, а затем полученное представление подается на вход генератору, который и выдает искомую маску. Этот метод демонстрирует высокую производительность внутри домена и выдающуюся обобщающую способность вне домена.

## 3 Выбор метода для исследования

Для исследования был выбран метод описанный в статье Label-Efficient Semantic Segmentation with Diffusion Models [1]. Авторы предлагают использовать предобученную модель типа DDPM, как backbone для получения новых представлений для пикселей сегментируемого изображения. Утверждается, что диффузионная модель хорошо выделяет семантические особенности изображений датасета, на котором она была обучена. Благодаря этому обучать классификаторы пикселей можно на маленьком количестве размеченных изображений, что безусловно является одним из главных преимуществ описанного метода сегментации. Также, предложенный в статье метод сегментации на базе DDPM сравнивается с другими, в том числе с методами на основе других генеративнных моделей(GAN, MAE).

# 4 Протокол тестирования и визуализации

## 4.1 Процесс обучения

В статье все методы сегментации обучались и тестировались на нескольких датасетах. Брался предобученный на большом датасете backbone(DDPM, GAN, MAE), а затем на маленькой, размеченной для сегментации части соответсвующего датасета обучался ансамбль классификаторов, который определяет класс пикселя по его представлению, полученному из backbone-а. Поступим так же.

## 4.2 Датасет

Для воспроизведения результатов и дальнейшего исследования были выбраны следующие датасеты для обучения классификаторов и тестирования метода сегментации:

- Horse-21:
  - содержит изображения лошадей
  - 21 семантических класса
  - 30 тренировочных изображений
  - 30 тестовых изображений
  - предполагается, что backbone был обучен на датасете LSUN-Horse
- FFHQ-34:
  - содержит изображения человеческих лиц
  - 34 семантических класса
  - 20 тренировочных изображений
  - 20 тестовых изображений
  - предполагается, что backbone был обучен на датасете FFHQ-256

- Cat-15:
  - содержит изображения кошек
  - 15 семантических классов
  - 30 тренировочных изображений
  - 20 тестовых изображений
  - предполагается, что backbone был обучен на датасете LSUN-Cat
- Bedroom-28:
  - содержит изображения спальных комнат
  - 28 семантических классов
  - 40 тренировочных изображений
  - 20 тестовых изображений
  - предполагается, что backbone был обучен на датасете LSUN-Bedroom

Были выбраны эти датасеты, так как для них доступны предобученные DDPM.

## 4.3 Метрики

Для оценки качества моделей было принято решение использовать ту же метрику, что и в исследуемой статье, а именно метрику mean-per-class-IoU, то есть среднее метрики Intersection-Over-Union по всем семантическим классам. Данная метрика является классической для задачи семантической сегментации и хорошо интерпретируется. IoU для истинной маски пикселей $A$ и предсказанной маски $B$ считается следующим образом:

$$\text{IoU}(A, B) := \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Для удобства во всех экспериментах метрика будет указана в процентах.

## 4.4 Визуализация

Для визуализации отрисовывается предсказанная сегментационная маска и отсматривается. Для сравнения также отрисовывается истинная маска.

# 5 Анализ и эксперименты

## 5.1 Запуск выбранного метода

Перед тем как приступать к экспериментам, были воспроизведены результаты исследуемой статьи. Для этого выбранный метод был запущен в исходном виде, предоставленном авторами статьи. В таблице 5.1 представлены результаты запусков.

| Method | Horse-21 | FFHQ-34 | Cat-15 | Bedroom-28 |
|---|---|---|---|---|
| init(my) | 63.74 | 58.03 | 57.34 | 49.49 |
| init(paper) | 65.0 ± 0.8 | 59.1 ± 1.4 | 53.7 ± 3.3 | 49.4 ± 1.9 |

Таблица 5.1: Сравнение методов сегментации по метрике mIoU

Метрики mIoU на датасетах FFHQ-34 и Bedroom-28 попадают в доверительные интервалы предоставленные авторами статьи. На датасете Horse-21 mIoU примерно на 0.5% меньше левой границы интервала, а на датасете Cat-15, наоборот, больше правой границы примерно на 0.3%. Вцелом можно считать, что результаты статьи были воспроизведены

Метрики, получаемые в дальнейших экспериментах, будем сравнивать с теми, которые получились у нас при запуске исходного метода.

## 5.2 Поиск слабых мест метода

Для нахождения слабых мест было проделано следующее:

- На тестовых изображениях каждого датасета был запущен исходный метод
- Для каждого отдельного изображения было посчитано mIoU
- В рамках каждого датасета тестовые изображения были отсортированы по возрастанию mIoU
- Вручную были отсмотрены изображения с худшими предсказаниями

Для начала посмотрим на изображения из каждого датасета с худшим mIoU. Они представлены на рисунке 5.1:

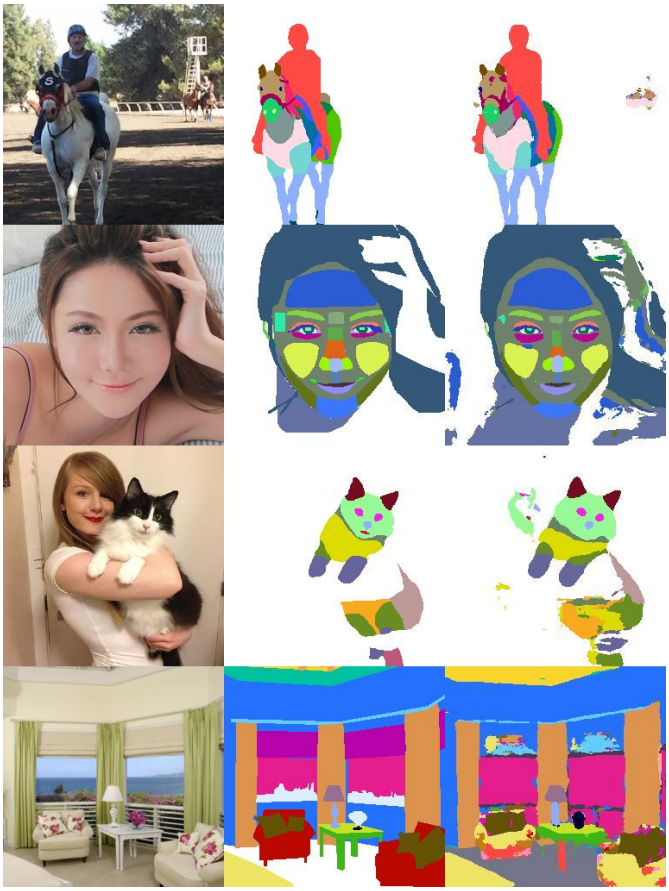


Рис. 5.1: Изображения, истинные маски, предсказанные маски

- В первом примере видим, что метод пытается сегментировать лошадь, которая находится на заднем фоне
- В примере из датасета FFHQ-34, видно, что основная проблема связана с тем, что метод плохо отличает лицо от других частей тела и поэтому пытается сегментировать руку, которая, на самом деле, должна оказаться частью фона
- В третьем примере похожая проблема: сегментируется не только кошка. То есть, проблема в том, что методу сложно определить где фон, а где объект, который непосредственно нужно сегментировать. Поэтому на фоне появляются шумы и артефакты
- В примере из датасета Bedroom-28 чуть другая проблема: из-за большого количества похожих форм одинакового цвета в комнате, метод сильно путается. Например можно заметить, что потолок, одеяло, и части кресел были определены в один класс

Рассмотрим на рисунке 5.2 другие примеры плохой работы метода, на которые стоит обратить внимание.

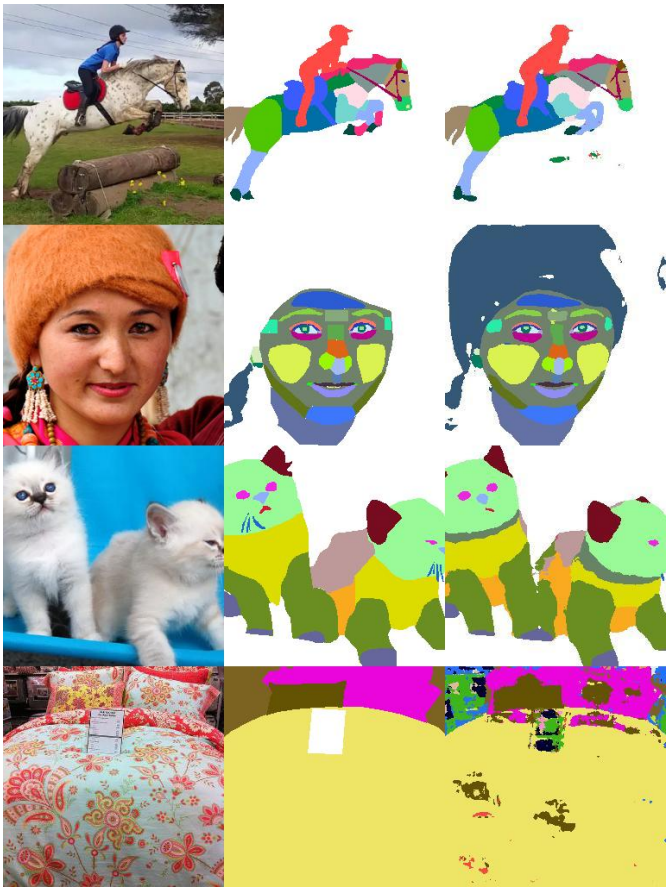


Рис. 5.2: Изображения, истинные маски, предсказанные маски

- В первом примере видим, что в общем и целом сегментация успешная, но есть некоторые шумы и артефакты там, где должен быть фон
- Во втором примере метод перепутал шапку с волосами
- В примере с котиками метод не заметил усы. Связано это с тем, что они сливаются по цвету с шерстью
- В четвертом примере из-за большого количества узоров на покрывале, метод выдал довольно шумную и неконсистентную сегментацию

Подведем некоторые итоги:

- Встречаются примеры, где метод пытается просегментировать что-то, что принадлежит фону. Из-за этого в результатах встречаются артефакты и шумные участки

- Как мы увидели, в некоторых примерах метод путал классы. Это может говорить о том, что возможно стоит подумать об улучшениях в способе классификации пикселей
- Также, отмечу, что есть dataset-specific проблемы, как, например, с шапками в датасете FFHQ-34. Но, так как семантическая сегментация применима в очень различных областях, то решать такие проблемы чревато потерей обобщающей способности метода

## 5.3 Выдвижение гипотез

### 5.3.1 Гипотеза 1

Как можно было заметить из примеров плохой работы метода, часто в сегментационной маске появляются шумы и артефакты на фоне. Для избавления от них, предлагается делать некоторую постобработку сегментационной маски. А именно воспользоваться морфологическими операциями dilation и erosion.

Морфологические операции — это математические операции, применяемые к изображению для извлечения определенных его характеристик. Две наиболее распространенные морфологические операции — это расширение (dilation) и сужение (erosion).

Операция расширения увеличивает область объекта на изображении. Она выполняется путем перемещения структурного элемента (обычно небольшого квадрата или круга) по изображению и замены каждого пикселя внутри структурного элемента на максимальное значение из пикселей, охваченных структурным элементом.

Аналогично, операция сужения уменьшает область объекта на изображении.

### 5.3.2 Гипотеза 2

В исходном методе ансамбль классификаторов обучается без каких либо аугментаций. Предлагается их добавить. Это потенциально может повысить обобщающую способность метода, а также аугментации могут помочь модели стать более устойчивой к шуму и отклонениям в данных.

Стоит учесть, что аугментации надо подбирать аккуратно, и они не должны быть слишком сложными. Хорошо подойдут pixel-level аугментации, которые не вносят много искажений и шума. Также можно применять различные геометрические аугментации, например повороты, обрезания. Важно учесть, что при применении таких аугментаций нужно их также применять к истинным маскам.

### 5.3.3 Гипотеза 3

Данная идея тоже связанна с аугментациями. Предлагается внедрить test-time аугментации.

Суть модификации в том, чтобы для очередного тестового изображения генерировать несколько аугментированных версий и для каждой из них предсказывать сегментационную маску, а затем путем голосования составлять итоговую маску.

Данная модификация позволит методу более уверенно относить пиксели к тому или иному классу. Также, потенциально шумных участков и артефактов на фоне должно стать меньше.

## 5.4 Проверка гипотез

### 5.4.1 Эксперимент 1

В этом эксперименте проверялась первая гипотеза, в которой предлагалось обрабатывать предсказанные маски с помощью морфологических операций erosion и dilation.

Для проверки гипотезы использовались маски полученные при запуске исходного метода. Над каждой предсказанной маской была проведена морфологическая операция закрытия (сначала erosion, затем dilation). Экспериментальным путем было выбрано ядро в виде креста размера 3x3. Результаты эксперимента можно увидеть в таблице 5.2.

| Method | Horse-21 | FFHQ-34 | Cat-15 | Bedroom-28 |
|---|---|---|---|---|
| init + post-processing | 63.6 | 57.16 | 56.86 | **49.68** |
| init | **63.74** | **58.03** | **57.34** | 49.49 |

Таблица 5.2: Сравнение методов сегментации по метрике mIoU

Метрика mIoU уменьшилась на всех датасетах, кроме Bedroom-28, на котором улучшение незначительное.

Посмотрим как работает постобработка на примере представленном на рисунке 5.3.

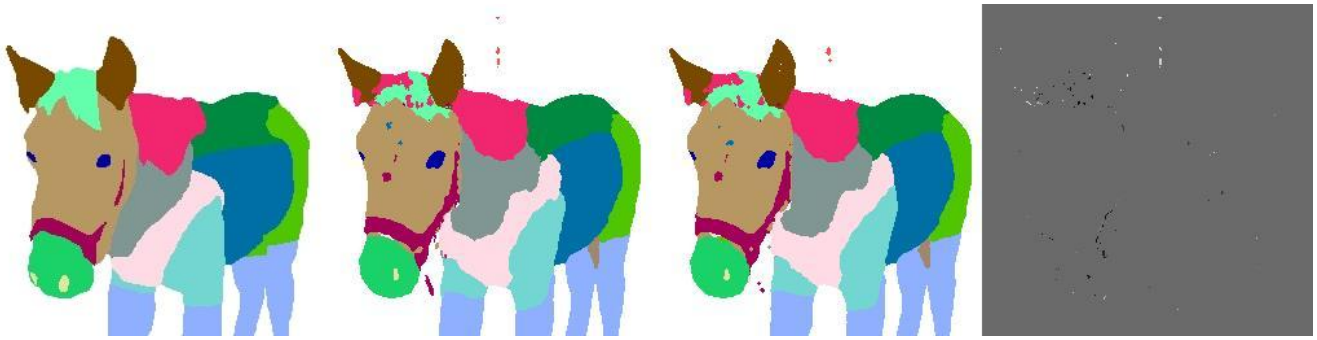


Рис. 5.3: Истинная маска, без постобработки, с постобработкой, разность предсказаний

Видно, что после морфологических операций, заметных глазу шумов становится меньше(в середине сверху и у морды). Но если посмотреть на дельту масок, то видно, что постобработка вносит изменения в других участках маски, поэтому и наблюдается небольшое уменьшение метрики вместо увеличения.

Вывод: количество убираемого таким путем шума меньше количества приобретаемого шума. Гипотеза отвергается.

### 5.4.2 Эксперимент 2

В этом эксперименте проверялась вторая гипотеза, в которой предлагалось использовать аугментации изображений при обучении классификаторов.

Применяемые аугментации указаны в таблице 5.3:

| **Augmentations** |
|---|
| ColorJitter |
| ToGray |
| HorizontalFlip |
| CoarseDropout |
| RandomResizedCrop |
| RandomRotation |
| ElasticTransform |

Таблица 5.3: Аугментации, использованные во время обучения

Каждая из аугментаций применялась с вероятностью 0.4, обучающие датасеты были увеличены в 5 раз (кроме обучающего датасета для Bedroom-28, он был увеличен в 4 раза).

Несмотря на увеличение датасетов, было решено количество итераций обучения за одну эпоху оставить прежним, чтобы сравнение с исходным методом было честным.

Результаты эксперимента в таблице 5.4.

| Method | Horse-21 | FFHQ-34 | Cat-15 | Bedroom-28 |
|---|---|---|---|---|
| init + augs | **65.14** | 57.32 | **59.43** | 49.03 |
| init | 63.74 | **58.03** | 57.34 | **49.49** |

Таблица 5.4: Сравнение методов сегментации по метрике mIoU

Отметим, что благодаря аугментациям на датасетах Horse-21 и Cat-15 есть значительные улучшения относительно воспроизведенных метрик. На датасетах FFHQ-34, Bedroom-28 метрика немного упала.

Гипотеза требует дальнейшего исследования.

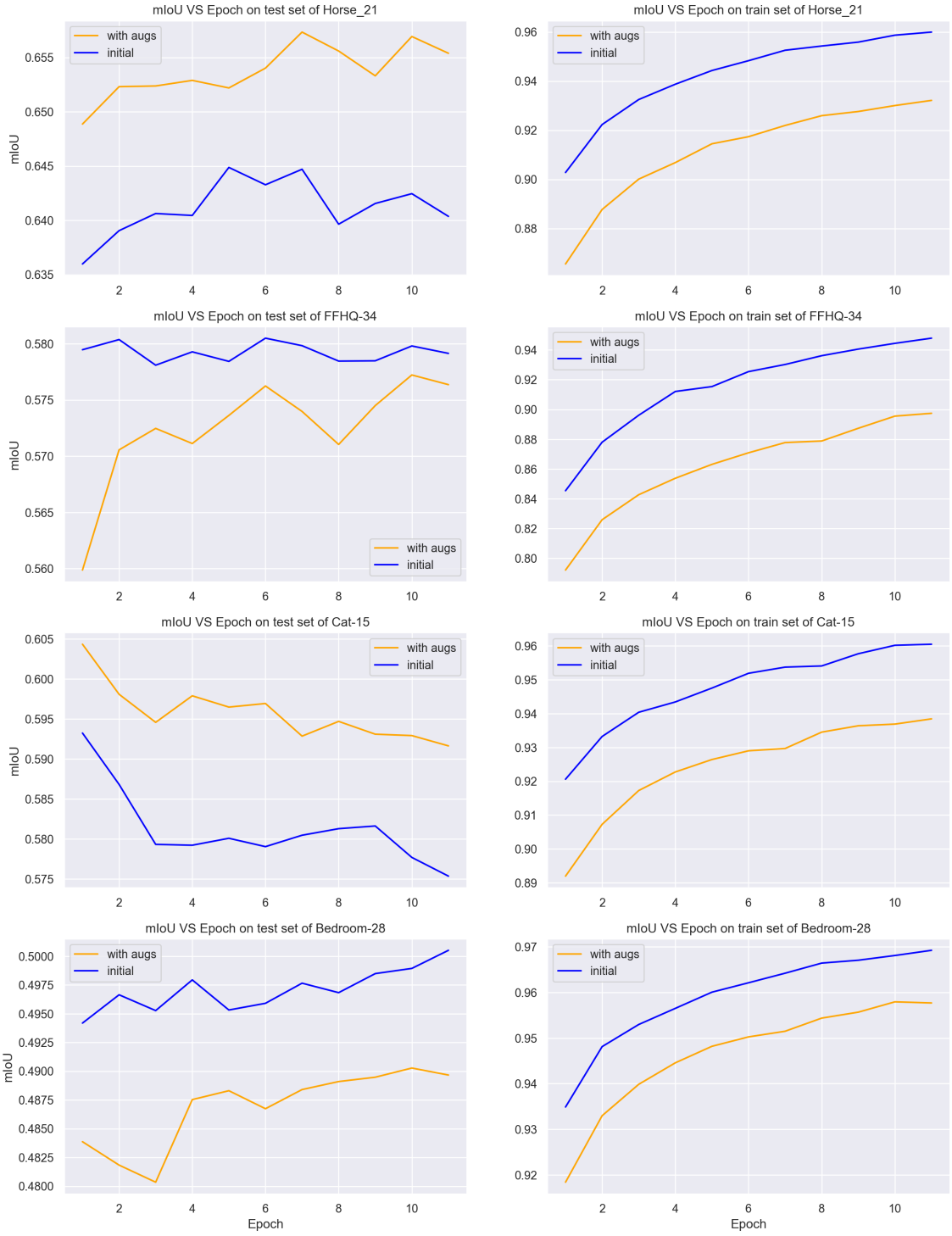
### 5.4.3 Эксперимент 3

После проведения предыдущего эксперимента возникла идея проверить деградируют ли метрики при увеличении количества эпох обучения. Было принято решение увеличить их количество до 11(было 4). Чтобы сравнение с исходным методом осталось валидным, в нем тоже количество эпох было увеличено. Результаты эксперимента можно увидеть в таблице 5.5

| Method | Horse-21 | FFHQ-34 | Cat-15 | Bedroom-28 |
|---|---|---|---|---|
| init + augs(11 epochs) | **65.53** | 57.63 | **59.16** | 48.96 |
| init(11 epochs) | 64.03 | **57.91** | 57.53 | **50.05** |

Таблица 5.5: Сравнение методов сегментации по метрике mIoU

Посмотрим также на график изменения mIoU на тестовых и тренировочных выборках в зависимости от номера эпохи.




Как можно заметить, после увеличения количества эпох с 4 до 11 метод с аугментациями, показал улучшения на датасетах Horse-21 и FFHQ-34, на датасете Bedroom-28 получилась примерно такая же метрика, а на Cat-15 она слегка просела. Если же сравнивать с исходным методом обученным так же на протяжении 11 эпох, ситуация остается такой же, что и в случае 4 эпох: на Horse-21 и Cat-15 модификация выигрывает, а на FFHQ-34 и Bedroom-28 проигрывает исходному методу.

Если посмотреть на графики mIoU на тестовых выборках, то понятно, что деградации метода не наблюдается, а на датасете FFHQ-34 и вовсе разность между метриками двух методов уменьшается. Также, стоит отметить, что на тренировочных данных mIoU модифицированного метода меньше чем у исходного, что может говорить о меньшей степени переобучения.

Дальше будем улучшать метод с аугментациями. Также, в дальнейших экспериментах будем использовать 11 эпох обучения для всех методов(в том числе и для исходного).

### 5.4.4 Эксперимент 4

В этом эксперименте мы попробовали объединить аугментации с другими методами борьбы с переобучением. Было сделано следующее:

- Оптимизатор Adam, который использовался авторами статьи, был заменен на AdamW с weight_decay=1e-4
- В архитектуру классификатора после каждого слоя ReLU были добавлены слои Dropout с вероятностью 0.4

Будем сравнивать эту модификацию, с иходным методом, а также с методом обученным с аугментациями. Метрики приведены в таблице 5.6.

| Method | Horse-21 | FFHQ-34 | Cat-15 | Bedroom-28 |
|---|---|---|---|---|
| init + augs + anti-overfit(11 epochs) | 65.2 | 57.37 | **59.53** | **50.39** |
| init + augs(11 epochs) | **65.53** | 57.63 | 59.16 | 48.96 |
| init(11 epochs) | 64.03 | **57.91** | 57.53 | 50.05 |

Таблица 5.6: Сравнение методов сегментации по метрике mIoU

Эта модификация слегка уступает методу использующему только аугментации на датасетах Horse-21 и FFHQ-34, но стоит заметить, что на Bedroom-28 прирост составляет примерно 1.4%. Если же делать сравнение с исходным методом, то данный метод выигрывает на трех датасетах, чего нельзя сказать про метод с аугментациями. По этим причинам дальнейшие эксперименты будут направлены на улучшение модификации обученной с аугментациями и борющейся с переобучением.

### 5.4.5 Эксперимент 5

В этом эксперименте проверялась третья гипотеза, в которой предлагалось внедрить test-time аугментации в исследуемый метод.

Test-time аугментация применялась следующим образом: для сегментируемого изображения генерировались аугментированные версии, для каждой из которых предсказывалась сегментационная маска, а итоговое предсказание делалось путем голосования.

Применялись 4 фиксированных набора аугментаций. Они перечислены в таблице 5.7:

| **Test-time augmentations** |
|---|
| ToGray, CourseDropout |
| ToGray |
| ColorJitter, HorizontalFlip |
| ColorJitter, HorizontalFlip, ChannelShuffle |

Таблица 5.7: Наборы аугментаций для применения в test-time

Test-time аугментации были внедрены в метод, полученный в предыдущем эксперименте. Метрики указаны в таблице 5.8.

| Method | Horse-21 | FFHQ-34 | Cat-15 | Bedroom-28 |
|---|---|---|---|---|
| init + anti-overfit + test-time augs(11 epochs) | **65.33** | **58.12** | **59.54** | **52** |
| init + anti-overfit(11 epochs) | 65.2 | 57.37 | 59.53 | 50.39 |
| init(11 epochs) | 64.03 | 57.91 | 57.53 | 50.05 |

Таблица 5.8: Сравнение методов сегментации по метрике mIoU

Гипотеза оказалась верной, так как использование test-time аугментаций привело к увеличению метрик на всех датасетах. Полученная модификация превосходит исходный метод на всех наборах данных. Будем считать этот метод итоговым.

# 6 Итоговый метод

## 6.1 Описание

За итоговый метод был взят тот, который получился в последнем эксперименте. Основные изменения, которые были применены к исходному методу для получения финального:

- Обучение на датасете с аугментациями
- Внедрены способы борьбы с переобучением: оптимизатор AdamW и Dropout слои в классификаторах
- Test-time аугментации

## 6.2 Количественные результаты

Метрики итогового и исходного методов представлены в таблице 6.1.

| Method | Horse-21 | FFHQ-34 | Cat-15 | Bedroom-28 |
|---|---|---|---|---|
| final | **65.33** | **58.12** | **59.54** | **52** |
| init | 64.03 | 57.91 | 57.53 | 50.05 |

Таблица 6.1: Сравнение исходного и итогового методов сегментации по метрике mIoU

На всех датасетах финальный метод показывает метрики лучше исходного. На датасетах Cat-15 и Bedroom-28 прирост составляет **2%**, на Horse-21 **1.3%**, а на FFHQ-34 **0.2%**.

## 6.3 Примеры работы

На рисунке 6.1 представлены примеры работы итогового метода, а также результаты исходного метода на тех же изображениях.

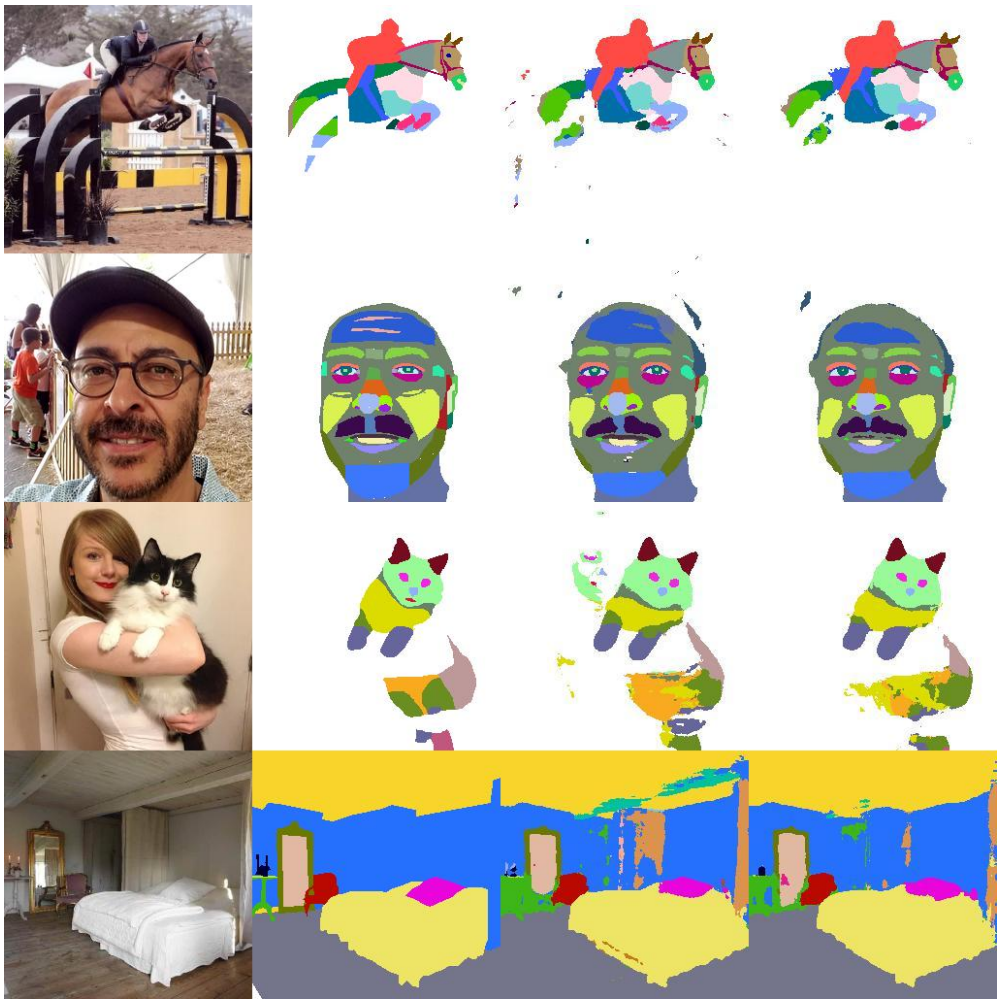


Рис. 6.1: Изображения, истинные маски, исходный метод, итоговый метод

Из примеров видно, что у итогового метода намного меньше шума и артефактов в предсказанных масках, особенно в фоновых участках. Таким образом, одно из главных слабых мест исходного метода отсутствует у итогового.

# 7 Выводы и результаты

В ходе проделанного исследования были изучены основные концепции использованные в генеративных диффузионных моделях. Также, был изучен и проанализирован SOTA метод использования диффузионных моделей для семантической сегментации изображений. На основе выдвинутых гипотез по улучшению были проведены эксперименты и был выведен итоговый метод, являющийся модификацией исследуемого подхода, основанный на использовании аугментаций данных и борьбе с переобучением. Полученный метод превзошел исследуемый метод на всех рассматриваемых датасетах.

# Список литературы


[1] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov и Artem Babenko. *Label-Efficient Semantic Segmentation with Diffusion Models*. 2022. arXiv: 2112.03126 [cs.CV].

[2] Jonathan Ho, Ajay Jain и Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. arXiv: 2006.11239 [cs.LG].

[3] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba и Sanja Fidler. *Semantic Segmentation with Generative Models: Semi-Supervised Learning and Strong Out-of-Domain Generalization*. 2021. arXiv: 2104.05833 [cs.CV].

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser и Björn Ommer. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2022. arXiv: 2112.10752 [cs.CV].

[5] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan и Surya Ganguli. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. 2015. arXiv: 1503.03585 [cs.LG].