

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет  
«Высшая школа экономики»**

**Факультет компьютерных наук  
Основная образовательная программа  
«Прикладная математика и информатика»**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ  
РАБОТА**  
**Исследовательский проект на тему**  
**Распознавания изображений, сгенерированных**  
**искусственным интеллектом.**

**Выполнил студент группы БПМИ218, 4 курса,  
Агеев Артем Андреевич**

**Руководитель ВКР:  
доцент ДБДИП ФКН НИУ ВШЭ, Максаев Артем Максимович**

**Куратор:  
руководитель группы развития технологий компьютерного зрения  
Ecom.tech, Савельев Александр Геннадьевич**

**Москва 2025**

# Содержание

<b>Аннотация</b>	<b>3</b>
<b>Annotation</b>	<b>3</b>
<b>Ключевые слова</b>	<b>3</b>
<b>1 Введение</b>	<b>4</b>
1.1 Описание предметной области . . . . .	4
1.2 Цели проекта . . . . .	5
<b>2 Анализ существующих решений</b>	<b>6</b>
2.1 Deep-Fake-Detector-v2 . . . . .	6
2.2 SuSy . . . . .	6
2.3 TruFor . . . . .	7
2.4 Сравнительный обзор и релевантность для задачи дорисовки изображений . . . . .	7
<b>3 Методология</b>	<b>9</b>
3.1 Общая архитектура . . . . .	9
3.2 Выбор генеративной модели . . . . .	9
3.3 Подбор оптимальных параметров . . . . .	10
3.3.1 Подбор алгоритма выборки . . . . .	11
3.3.2 Подбор запроса . . . . .	13
3.3.3 Подбор количества шагов . . . . .	13
3.4 Поиск зоны для подмены . . . . .	14
3.5 Верификация подмены . . . . .	14
3.6 Классификация изображений . . . . .	15
<b>4 Подготовка данных</b>	<b>16</b>
<b>5 Результаты</b>	<b>17</b>
<b>6 Заключение</b>	<b>19</b>
<b>Список литературы</b>	<b>20</b>

## Аннотация

Последние достижения в области генеративных моделей искусственного интеллекта существенно расширили возможности злоумышленников по созданию фальсифицированных изображений, содержащих высококачественный текст. Это создает серьёзную угрозу безопасности, так как упрощает изготовление поддельных фотографий документов. Особенно актуально применение методов дорисовки (inpainting), позволяющих заменять текстовые элементы на реальных изображениях. В настоящей работе проводится анализ моделей, применимых для решения данной задачи, и предлагается автоматизированный процесс, обеспечивающий замену фрагментов изображения содержащих текст, с визуальным качеством, достаточным для введения в заблуждение как человеческих наблюдателей, так и современных систем обнаружения на основе искусственного интеллекта. Предложенный процесс также используется для генерации датасета, предназначенного для обучения модели, способной выявлять изображения с подобной подменой. Представленная модель также сравнивается с другими детекторами генеративного ИИ. Исходный код доступен по ссылке: [\[1\]](#).

## Annotation

Recent advancements in generative AI models have allowed malicious actors to generate fake images with high-quality text. This poses a significant risk by enabling the creation of counterfeit document photographs. In particular, inpainting can be used to replace patches containing text on real images. This paper examines the models suitable for such inpainting and proposes a pipeline to automatically replace parts containing text in images with output quality high enough to deceive both human observers and existing AI detectors. This pipeline is then utilized to generate a dataset for training a model capable of identifying such inpainted images. The model is then compared to existing generative AI detectors. Source code available at: [\[1\]](#).

## Ключевые слова

Inpainting, Generative AI, AI detectors, Дорисовка, Генеративный ИИ, Детекторы ИИ

# 1 Введение

## 1.1 Описание предметной области

В последние годы, достижения в области генеративного искусственного интеллекта позволили создавать фотореалистичные изображения [22]. Современные модели генерируют изображения с такой степенью реалистичности, что они практически неотличимы от настоящих фотографий для нетренированного наблюдателя [16]. Это вызывает серьёзные опасения относительно возможного злоупотребления данной технологией для создания дипфейков. В результате, возникло новое направление исследований, посвящённое методам выявления поддельных изображений, созданных с использованием инструментов ИИ.

Однако, современные модели, такие как FLUX 1.1 [15] и Stable Diffusion 3.5 [2], представляют собой новую угрозу, поскольку способны генерировать реалистичный текст точно в соответствии с заданным описанием. Модели предыдущих поколений не были способны на это, ведь текст часто был сильно искажен, а также были пропуски букв или целых слов [1.1]. Это позволяет злоумышленникам создавать поддельные изображения, которые ранее было трудно или невозможно сгенерировать методами искусственного интеллекта, например фасады магазинов или фотографии товаров.



Рис. 1.1: Сравнение диффузионных моделей разных поколений. Запрос: “A shop with a large billboard saying ‘Welcome’”

Более того, при использовании этих моделей в режиме дорисовки (inpainting) появляется возможность замены конкретных элементов настоящих фотографий. При таком использовании преимуществом генеративных моделей по сравнению с традиционными средства-

ми редактирования изображений является способность модели не только изменить текст на участке изображения, но и создать правдоподобный фон, а также учитывать перспективу и общую стилистику. Подобные модификации может быть сложно и трудозатратно делать традиционными средствами редактирования изображений, особенно в автоматическом режиме, однако генеративные модели способны создать естественный переход от текста к окружающей области за пределами зоны дорисовки. Это делает подмену намного более правдоподобной и затрудняет её выявление даже при внимательном визуальном анализе со стороны человека. Однако существующие детекторы ИИ также сталкиваются со сложностями в выявлении таких изображений, ведь значительная часть изображения является действительно настоящей и может нивелировать детектируемые сигналы вмешательства, особенно в случаях, когда дорисованная область составляет лишь небольшую часть изображения.

## 1.2 Цели проекта

Целью данного проекта является создание автоматизированного процесса по замене фрагментов изображений содержащих текст. Данный процесс позволит создать датасет фото с настоящим и подменённым текстом, на котором возможно обучить детектор, успешно обнаруживающий модификацию изображений.

Основные требования к полученным изображениям:

- Текст, полученный в результате замены должен совпадать с текстом в запросе модели.
- Полученное изображение должно обладать фотoreалистичным качеством:
  - не допускается размытый фон
  - текст на изображении должен быть четким и легко читаемым
  - если заменяемый текст является частью большего текста, то полученный текст должен повторять стилистику текста вокруг
- Полученные изображения должны демонстрировать устойчивость к обнаружению существующими детекторами.

## 2 Анализ существующих решений

Развитие методов детектирования синтетических изображений в последние годы проходит столь же стремительно, как и совершенствование самих генеративных моделей. Ниже рассматриваются два представителя современного класса открытых решений: **Deep-Fake-Detector-v2** [18] от prithivMLmods и **SuSy** [10][4] от HPAI-BSC.

### 2.1 Deep-Fake-Detector-v2

Deep-Fake-Detector-v2 базируется на Vision Transformer `google/vit-base-patch16-224-in21k` (85,8 М параметров) и решает бинарную классификацию “Realism/Deepfake”. Модель дообучена две эпохи на расширенной подборке реальных и синтетических лицевых изображений с интенсивными аугментациями. На валидации заявлены точность 0.921 и сбалансированная  $F_1 = 0.92$ . Использование трансформерной архитектуры позволяет учитывать глобальный контекст изображения, что потенциально полезно при анализе высокоуровневых артефактов, однако данный детектор не считается эффективным для обнаружения модифицированных изображений. Тем не менее, данная модель была включена в сравнительный анализ в качестве представителя универсальных детекторов, не специализированных на задачах обнаружения дорисованных изображений.

### 2.2 SuSy

SuSy (Spatial-based Synthetic image detector and attribution model) решает задачу как бинарной идентификации подлинности, так и мультиклассовой атрибуции изображения к одной из шести категорий (пять генеративных моделей + “реальное” изображение). Архитектурно модель представляет собой компактную CNN с каскадом “бутылочных горлышек” на основе ResNet-18: экстрактор признаков (12,5 М параметров) формирует многоуровневое пространственное представление, которое агрегируется адаптивным усреднением и подаётся в трёхслойный MLP-классификатор (ещё 0,2 М параметров). Входом служат фрагменты изображения размером  $224 \times 224$  пикселей. Для получения результата на изображении используется оконный обход и голосование по фрагментам.

Модель обучена на собственном датасете, объединяющем  $\approx 18$ тыс. изображений из COCO (реальные) и диффузионных генераторов различных поколений — Stable Diffusion 1.3/1.4/2, Midjourney V5, DALL·E 3 и др. На тесте авторов SuSy достигает полноты 0.965 для изображений, сгенерированных Flux 1-dev (2024), и 0.899 на “in-the-wild” наборе неизвестных

источников, демонстрируя высокую обобщающую способность.

Благодаря явной обработке фрагментов изображений, данный классификатор значительно лучше подходит для задачи обнаружения модифицированных изображений.

### 2.3 TruFor

TruFor [8] (Trustworthy Image Forgery Detection and Localization) — универсальный детектор, ориентированный на обнаружение локальных манипуляций как “классических” (copy-move, сплайсинг), так и современных, выполненных диффузионными моделями. Ключевая идея — одновременный анализ высокоуровневых RGB-признаков и низкоуровневого отпечатка Noiseprint++ обучаемого в самостоятельном режиме “шумового” представления, чувствительного к артефактам, вносимым камерой и постобработкой. Две ветви (RGB и Noiseprint++) подаются в cross-modal блок на основе трансформера, после чего система выдаёт сразу три карты:

- Anomaly map — побитовая локализация предполагаемых подмен (формулируется как задача бинарной сегментации);
- Confidence map — оценка надёжности каждой области; помогает подавлять ложноположительные вспышки “шума”;
- Integrity score — глобальная вероятность подделки для всего изображения, вычисляемая с учётом двух предыдущих карт.

Такой дизайн позволяет одновременно решать и задачу детекции, и задачу локализации, причём с повышенной устойчивостью к зашумлению изображения (изменение параметров сжатия, изменение размера). В экспериментальном сравнении на восьми открытых наборах (CASIA v1, Coverage, Columbia, NIST-16, DSO-1, VIPP, OpenForensics, CocoGlide) TruFor показал наивысшую среднюю  $F_1$ -метрику по локализации (0.785 при фиксированном пороге 0.5) и баланс-accuracy  $\approx 0.78$  на уровне всей картинки, заметно опередив CAT-Net v2, MVSS-Net и IF-OSN.

### 2.4 Сравнительный обзор и релевантность для задачи дорисовки изображений

Так как Deep-Fake-Detector-v2 обрабатывает изображение целиком, обнаружение им дорисовки изображений маловероятно, так как настоящая часть изображения будет мас-

кировать подмененную. Напротив, SuSy и TruFor обрабатывают части изображений (SuSy используя фрагменты изображений, TruFor используя карты изображения), что позволяет им успешно обнаруживать дорисованные изображения.

Таблица 2.1: Сравнение существующих детекторов

	<b>Deep-Fake-Detector-v2</b>	<b>SuSy</b>	<b>TruFor</b>
Архитектура	ViT-Base (85.8 M)	CNN (12.7 M)	SegFormer (43 M)
Тип вывода	2 класса (Real/Deepfake)	6 классов (5 ИИ + Real)	2 класса + карта
Обнаружение дорисовки	Нет	Да	Да

## 3 Методология

### 3.1 Общая архитектура

Так как в рамках данной работы рассматривается не создание полноценного изображения, а подмена текста на существующем изображении, важной частью является поиск зоны для подмены. Благодаря тому, что нам не требуется знать какой именно текст мы подменяем, возможно использовать специализированную нейросеть поиску областей, содержащих текст. Наиболее вероятная область текста используется как зона для дорисовки изображения. Подробное описание этой части алгоритма находится в разделе 2.4.

Одной из ключевых проблем применения генеративных моделей искусственного интеллекта является ограниченная управляемость процессом генерации, что затрудняет получение изображений строго соответствующих заданному описанию. В случае работы с текстом эта проблема усугубляется, так как модели часто могут пропускать отдельные буквы или целые слова, особенно когда текст длинный и есть незнакомые им слова. Так как основные требования к алгоритму требуют гарантий на корректную замену текста, необходим способ подтверждения, что генеративная модель действительно составила корректный текст. Для этого используется метод оптического распознавания символов в зоне подмены текста. Подробное описание этой части алгоритма находится в разделе 2.5.

Таким образом, общий алгоритм представляется следующей блок-схемой:



Рис. 3.1: Этапы обработки изображения.

### 3.2 Выбор генеративной модели

В качестве основной модели для подмены текста используется FLUX.1-dev [15] от группы black-forest-labs. В частности используется FLUX-Controlnet-Inpainting [20] авторства группы Alimama Creative. Использование модели Flux напрямую затрудняется тем, что модель зачастую игнорирует запрос и лишь дорисовывает область замены подстраиваясь под окружение. Использование модели на основе Controlnet позволяет модели лучше следовать

требованиям запроса, в частности если область дорисовки частично задевает соседние слова в тексте, модель лучше справляется с тем, чтобы приоритезировать создание требуемого текста по сравнению с дорисовкой существующего. Эта крайне важная черта, ведь поиск зоны для подмены не идеален и на сложных фотографиях, где есть большое количество маленького текста, часто может добавлять в область замены части соседнего текста.

Также в качестве базовой модели рассматривалась модель Stable Diffusion XL [3], в частности интеграция на основе Controlnet: SD-XL Inpainting 0.1 [21]. Другие модели с открытыми весами не рассматривались, так как уступают представленным моделям в генерации текста даже не в режиме дорисовки. Однако в сравнении решений FLUX-Controlnet-Inpainting и SD-XL Inpainting 0.1, предпочтение было отдано первой модели, во многом благодаря её способности лучше следовать требованиям к тексту в запросе к модели. Стоит отметить, что общий подход к модификации текста на изображениях, описанный в этой статье, не опирается на использование лишь одной модели и может использоваться с несколькими моделями одновременно для создания более общего датасета, что улучшает способность полученного детектора обнаруживать подмену текста, выполненную с помощью незнакомых моделей.

### 3.3 Подбор оптимальных параметров

Для подбора оптимальных параметров использовался набор тестовых изображений с достаточно простым фоном, а также созданной вручную областью подмены.

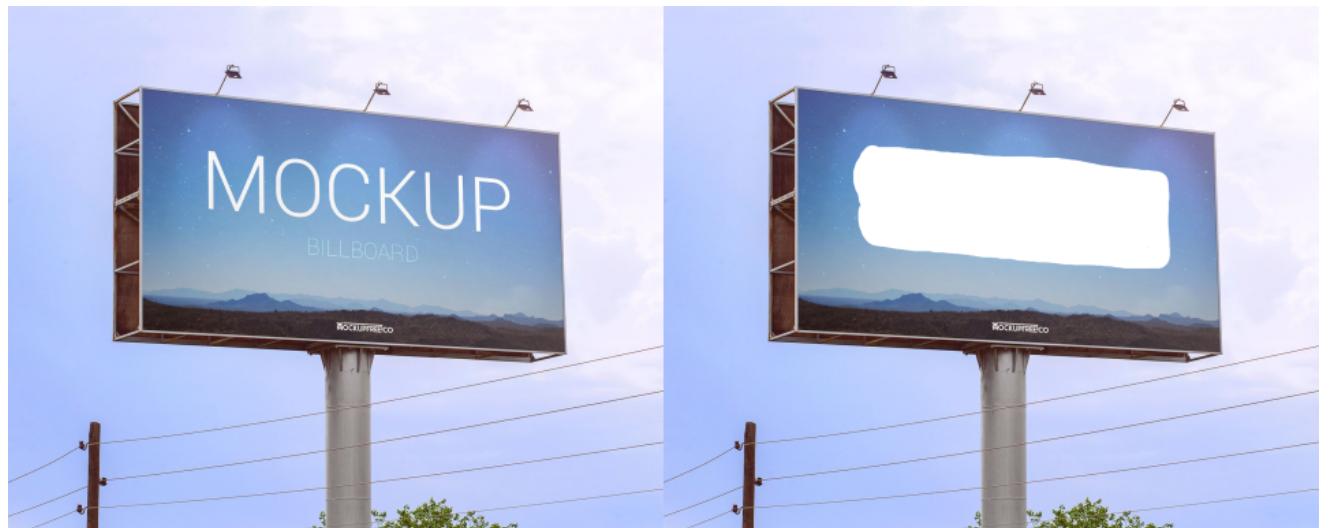


Рис. 3.2: Пример тестового изображения для подбора параметров: оригинал и с областью дорисовки.

При выборе тестовых изображений использовались следующие требования:

- Сложный, неоднородный фон с градиентом - используется для тестирования способности модели подстраиваться под окружение области дорисовки.
- Наличие перспективы - проверяет способность модели учитывать расположение текста в пространстве.

Для запуска модели использовалась модификация графа обработки для ComfyUI [5] представленного авторами FLUX-Controlnet-Inpainting. Ввиду ограничений на объем доступной видеопамяти использовалась версия Flux1-dev, использующая 8-битное представление чисел (FP8). Для кодирования запросов к модели используется T5-XXL [7], использующая 16-битное представление чисел (FP-16).

Список подбираемых параметров в приблизительном порядке приоритета:

1. Sampler - Алгоритм выборки изображений из латентного пространства
2. Positive prompt - Запрос для модели, описывает требования к генерации
3. Steps - Количество шагов процесса восстановления изображения

Другие фиксированные параметры:

- CLIP guidance (3.5) - направляющий коэффициент CLIP-кодировщика [19]
- Разрешение изображения (1024x1024)
- CFGGuider coefficient (1.0) - коэффициент CFG-направляющей
- Negative prompt (bad, ugly, deformed) - Отрицательный запрос для модели

Данные параметры были указаны как оптимальные группой Alimama Creative, и поэтому не подбирались.

### 3.3.1 Подбор алгоритма выборки

При выборе алгоритма выборки хорошо себя показали следующие алгоритмы:

- Euler [13] - Алгоритм по умолчанию, основанный на методе Эйлера.
- Euler ancestral [12] - Версия метода Эйлера с “наследованием шума”.
- DPM++ 2s ancestral [11] - Дифференциальный метод второго порядка с “наследованием шума”.
- LCM [14] - Модель латентной согласованности



Рис. 3.3: Euler



Рис. 3.4: Euler ancestral



Рис. 3.5: DPM++ 2s ancestral



Рис. 3.6: LCM

Рис. 3.7: Примеры методов выборки. Запрос: “A billboard with a blue-purple night sky filled with stars saying: “A complicated text to inpaint!””

В качестве итогового алгоритма выборки был выбран LCM, поскольку при его использовании наблюдалось существенно меньшее количество артефактов в области текста по сравнению с альтернативными методами. Стоит отметить, что данный метод проигрывает остальным в генерации фона за текстом, часто делая его несколько размытым или добавляя повторяющиеся структуры. Однако при необходимости данные недостатки возможно компенсировать предоставляя модели в запросе детальное описание фона за текстом, а также подбору ‘удачного’ стартового значения генератора чисел. В случае других методов повысить качество генерации непосредственно текста не удаётся.

### 3.3.2 Подбор запроса

В рамках тестирования выяснилось, что запрос модели очень сильно влияет на качество генерируемого текста. Так как общая схема не предусматривает подбор оптимального запроса для каждого изображения индивидуально, то требуется найти общий запрос, с которым модель стабильно выдает изображения хорошего качества.

Изначально предполагалось использовать максимально абстрактный запрос, например “Text saying: ‘Hello!’” (Текст говорящий: ‘Привет!’). Преимуществом такого запроса является его общность, позволяющая использовать его для подмены текста на любых изображениях. Однако эта особенность является и главным недостатком, так как ввиду своей общности, модель зачастую могла полностью игнорировать запрос и не создавать запрашиваемый текст, а лишь дорисовывать фон под окружение области подмены.

Намного лучше себя зарекомендовали запросы, в которых, помимо непосредственно команды добавить требуемый текст, есть описание самой картинки, например: “A billboard saying ‘Hello!’ with a blue-and-purple night sky filled with stars” (Рекламный щит с надписью “Привет!” с голубо-фиолетовым ночным небом, усеянным звездами). За частую такое детальное описание позволяет модели создавать намного более естественный фон, лучше сливающийся с немодифицируемой частью изображения.

Также стоит отметить, что на примерах с более сложным фоном модель чаще создает правильный текст, если добавлять описание самого текста, например: “A text saying ‘Hello!’ in clear, large and white letters.” (Текст ‘Привет!’ написанный четкими, большими, белыми буквами).

### 3.3.3 Подбор количества шагов

Оптимальное количество шагов для всех алгоритмов выборки, кроме LCM, находится в пределах от 30 до 50. В случае LCM, оптимальное количество шагов находится в пределах от 20 до 50, однако стоит отметить, что ввиду особенностей работы LCM, близкие друг к другу шаги могут производить сильно разные изображения. Из-за этого в случае неудачного результата, остальные алгоритмы требуют выбора другого стартового значения генератора случайных чисел, в то время как при использовании LCM есть возможность декодировать промежуточные состояния, так как с большой вероятностью среди них будут хорошие изображения. Это позволяет значительно повысить долю успешно созданных изображений. Впрочем, на сложных примерах и LCM может потребовать выбора другого стартового значения.

### **3.4 Поиск зоны для подмены**

Для определения зоны, где будет производиться дорисовка изображения используется отдельное приложение, использующее детектор текста EAST [23]. Алгоритм обработки следующий:

1. Обрезание и масштабирование изображения до разрешения 1024 на 1024 пикселя
2. Создание набора окон - прямоугольников вида  $[(800, 800), (640, 640), (800, 480), (800, 320)]$  пикселей.
3. Для каждого окна производится сканирование изображения со сдвигом 32 пикселя
4. Среди всех окон и сдвигов выбирается лучшее окно (лучший результат детектора текста)

Область внутри окна заполняется прозрачными пикселями и становится областью для дорисовки.

### **3.5 Верификация подмены**

Для проверки, что полученное изображение действительно содержит требуемый текст, используется оптическое распознавание символов (OCR) на основе Tesseract [6]. В обработке участвует только часть изображения, где происходила замена текста. Для улучшения качества распознавания текста используется следующая предобработка:

1. Поворот изображения на угол от  $-15$  до  $+15$  градусов (перебираются все повороты с шагом в 1 градус).
2. Увеличение разрешения изображения в 3.5 раза с применением бикубической интерполяции. Данный шаг требуется для более корректной работы морфологической фильтрации
3. Локальная адаптивная бинаризация с использованием метода Гауссовой пороговой фильтрации (adaptive thresholding). Для каждого пикселя порог рассчитывается на основе взвешенного среднего в окрестности (размером 201 пикселей).
4. Морфологическая фильтрация: сначала выполняется операция открытия (удаление мелких шумов), затем закрытие (заполнение мелких разрывов внутри символов). В обоих случаях используется структурирующий элемент размером  $8 \times 8$  пикселей. Данный шаг значительно снижает шум в изображении.

5. Распознавание текста с помощью Tesseract в режиме построчного анализа. В итоговый результат включаются только те фрагменты текста, для которых уровень уверенности составляет не менее 80%.
6. Для каждого результата распознавания (всего 31 вариант, соответствующий всем возможным поворотам текста) вычисляется метрика символьного сходства между эталонной строкой и результатом OCR. В качестве метрики сходства используется нормированная наибольшая общая подпоследовательность (число от 0 до 1).

Среди всех углов поворота изображения выбирается результат с наибольшим сходством к целевому тексту. В большинстве случаев при правильной генерации текста, результат будет строго 1. Таким образом, при необходимости требовать высочайшее качество генерации возможно использовать отсечку 1, однако в данной работе использовалась отсечка 0.8, ввиду редкого ложноположительного обнаружения символов. Также это позволяет допускать случаи дописывания моделью соседнего с областью дорисовки текста.

Данный подход позволяет хорошо распознавать даже сильно стилизованный текст на сложных фонах. Несмотря на то, что области дорисовки изображения строго горизонтальные, поворот текста используется для лучшего обнаружения текста, искаженного перспективой.

### 3.6 Классификация изображений

Для определения является ли изображение настоящим или содержит сгенерированный текст использовался бинарный классификатор на основе ResNet-50 [9]. Полученное изображение обрезается до разрешения 224x224 пикселей, нормализуется по средним значениям и стандартным отклонениям ImageNet и подается на вход модели ResNet. В качестве финального слоя модели используется линейный слой с 2 выходами. Такая простая архитектура обуславливается тем, что основной акцент в данном исследовании сделан именно на создании и качестве датасета, а не на усложнении или оптимизации архитектуры модели-детектора.

## 4 Подготовка данных

В качестве основы для составления датасета был выбран датасет Commercial Façades Dataset[17]. В нем представлены 4000 фотографий в категориях: магазины, рестораны, отели и прочее. Категория прочее не использовалось, и часть фотографий в других категориях не удалось загрузить. В конечном счете использовалось 2743 фотографии, в частности: 961 магазин, 970 ресторанов, 812 отелей. Каждая из фотографий была обработана алгоритмом, описанным в секции 3. Для части фотографий не удалось составить текст даже спустя несколько попыток. В результате получился датасет из 2743 оригинальных фотографий и 1184 изображений с успешной заменой текста, а также 1559 изображений без успешной замены текста. Изначально планировалось обучать детектор только на изображениях с успешной заменой текста, однако так как все изображения содержали одинаковый текст 'Hello!', это вызвало опасения, что модель будет обучаться распознавать наличие этого общего текста. Когда данные опасения подтвердились, было принято решение использовать все изображения, включая неуспешную замену текста. Также было создано 300 изображений с другим текстом 'Astronomia', на которых было подтверждено, что качество детектора не падает при изменении текста.

## 5 Результаты

Таблица 5.1: Метрики для Deep-Fake-Detector-v2

	Precision	Recall	F1-score	Support
Fake	0.310	0.005	0.009	2743
Real	0.499	0.989	0.663	2743
Accuracy			0.497	5486
Macro avg	0.404	0.497	0.336	5486
Weighted avg	0.404	0.497	0.336	5486

Как и предполагалось, Deep-Fake-Detector-v2 не способен обнаруживать изображения с подменой текста. Из 1893 изображений в тестовой выборке лишь 10 были корректно классифицированы как ‘Fake’. Это демонстрирует, что предложенный метод генерации изображений крайне эффективен в обходе детекторов общего назначения, в задачи которых не входит поиск изображений с дорисовкой.

Таблица 5.2: Метрики для SuSy

	Precision	Recall	F1-score	Support
Fake	0.719	0.695	0.707	2743
Real	0.705	0.729	0.717	2743
Accuracy			0.712	5486
Macro avg	0.712	0.712	0.712	5486

Детектор SuSy показал себя несколько лучше. Стоит отметить, что в рамках тестирования использовалось 16 патчей, покрывающих изображение целиком. В данном случае порог настоящего изображение был изменен с 0.5 до 0.7. В таком случае модель имеет больше ложноположительных ошибок, однако общая  $F_1$  мера в таком случае максимальна и составляет 0.712.

Таблица 5.3: Метрики для TruFor

	Precision	Recall	F1-score	Support
Fake	0.737	0.203	0.318	2743
Real	0.538	0.928	0.681	2743
Accuracy			0.565	5486
Macro avg	0.638	0.565	0.499	5486

Несмотря на высокую способность архитектуры TruFor к анализу мельчайших деталей изображений, её эффективность может быть ограничена недостаточным количеством

обучающих данных, сгенерированных при помощи диффузионных моделей. Кроме того, доступные синтетические изображения, использованные при обучении, утратили актуальность в условиях стремительного развития генеративных моделей. В результате модель имеет высокую полноту на настоящих изображениях, однако полнота на модифицированных изображениях составляет лишь 20%.

Таблица 5.4: Метрики для предложенного детектора

	Precision	Recall	F1-score	Support
Real	0.958	0.958	0.958	571
Fake	0.954	0.954	0.954	527
Accuracy			0.956	1098
Macro avg	0.956	0.956	0.956	1098
Weighted avg	0.956	0.956	0.956	1098

Предложенный в данной статье детектор показал значительно лучшие результаты по всем основным метрикам. Такая высокая точность позволяет крайне эффективно использовать предложенный классификатор для фильтрации изображений в автоматизированных системах, где важна высокая надежность.

## 6 Заключение

В ходе данной работы была представлена автоматизированная система по замене фрагментов изображения содержащих текст. Данная система способна обнаружить области содержащие текст, заменить их при помощи диффузионной модели и подтвердить качество замены при помощи оптического распознавания символов. Дорисовка изображения обладает высоким качеством и с трудом обнаруживается как существующими автоматическими детекторами, так и людьми.

С использованием этой системы был составлен набор данных на основе фотографий фасадов магазинов, отелей и ресторанов, состоящий из настоящих и модифицированных изображений. На данном наборе данных были протестированы существующие детекторы синтетических изображений Deep-Fake-Detector-v2, SuSy и TruFor. Также был обучен и протестирован авторский детектор на основе ResNet.

В рамках тестирования, существующие методы не могли эффективно отличить модифицированные изображения от оригиналов, в то время как представленный детектор обладал высокой точностью и полнотой.

## Список литературы

- [1] Artem Ageev. *Code and images*. <https://github.com/codereptile/bachelor-thesis>. 2025.
- [2] Stability AI. *Stable Diffusion 3.5*. <https://huggingface.co/stabilityai/stable-diffusion-3.5-large>. Accessed: 2025-05-14. 2024.
- [3] Stability AI. *Stable Diffusion XL Base 1.0*. <https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0>. Accessed: 2025-05-14. 2023.
- [4] Pablo Bernabeu-Perez, Enrique Lopez-Cuena и Dario Garcia-Gasulla. *Present and Future Generalization of Synthetic Image Detectors*. 2024. arXiv: [2409.14128 \[cs.CV\]](https://arxiv.org/abs/2409.14128). URL: <https://arxiv.org/abs/2409.14128>.
- [5] comfyanonymous. *ComfyUI*. <https://github.com/comfyanonymous/ComfyUI>. Accessed: 2025-05-14. 2023.
- [6] Tesseract OCR Developers. *Tesseract Open Source OCR Engine*. <https://github.com/tesseract-ocr/tesseract>. Accessed: 2025-05-14. 2025.
- [7] Google. *FLAN-T5 XXL*. <https://huggingface.co/google/flan-t5-xxl>. Accessed: 2025-05-14. 2022.
- [8] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour и Luisa Verdoliva. «TruFor: Leveraging all-round clues for trustworthy image forgery detection and localization». B: *arXiv preprint arXiv:2212.10957* (2022). Accessed: 2025-05-14. URL: <https://arxiv.org/abs/2212.10957>.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren и Jian Sun. «Deep Residual Learning for Image Recognition». B: *CoRR* abs/1512.03385 (2015). Accessed: 2025-05-14. arXiv: [1512.03385](https://arxiv.org/abs/1512.03385). URL: [http://arxiv.org/abs/1512.03385](https://arxiv.org/abs/1512.03385).
- [10] HPAI-BSC. *SuSy - Synthetic Image Detector*. <https://huggingface.co/HPAI-BSC/SuSy>. Accessed: 2025-05-14. 2024.
- [11] Hugging Face. *DDPM Scheduler – Diffusers v0.11.0 Documentation*. <https://huggingface.co/docs/diffusers/v0.11.0/en/api/schedulers/ddpm>. Accessed: 2025-05-14. 2022.
- [12] Hugging Face. *Euler Ancestral Scheduler – Diffusers v0.11.0 Documentation*. [https://huggingface.co/docs/diffusers/v0.11.0/en/api/schedulers/euler\\_ancestral](https://huggingface.co/docs/diffusers/v0.11.0/en/api/schedulers/euler_ancestral). Accessed: 2025-05-14. 2022.

- [13] Hugging Face. *Euler Scheduler – Diffusers v0.11.0 Documentation*. <https://huggingface.co/docs/diffusers/v0.11.0/en/api/schedulers/euler>. Accessed: 2025-05-14. 2022.
- [14] Hugging Face. *Latent Consistency Model Multistep Scheduler – Diffusers Documentation*. <https://huggingface.co/docs/diffusers/api/schedulers/lcm>. Accessed: 2025-05-14. 2024.
- [15] Black Forest Labs. *FLUX.1-dev*. <https://huggingface.co/black-forest-labs/FLUX.1-dev>. Accessed: 2025-05-14. 2024.
- [16] Hanno Labuschagne. *AI-generated images of celebrities are fooling people*. <https://mybroadband.co.za/news/software/485573-ai-generated-images-of-famous-people-are-fooling-people.html>. Accessed: 2025-05-16. 2023.
- [17] madrugado. *Commercial Façades Dataset*. <https://github.com/madrugado/commercial-facades-dataset/>. Accessed: 2025-05-14. 2025.
- [18] prithivMLmods. *Deep-Fake-Detector-v2-Model*. <https://huggingface.co/prithivMLmods/Deep-Fake-Detector-v2-Model>. Accessed: 2025-05-14. 2024.
- [19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger и Ilya Sutskever. «Learning Transferable Visual Models From Natural Language Supervision». B: *arXiv preprint arXiv:2103.00020* (2021). Accessed: 2025-05-14. URL: <https://arxiv.org/abs/2103.00020>.
- [20] Alimama Creative Team. *FLUX-Controlnet-Inpainting*. <https://github.com/alimama-creative/FLUX-Controlnet-Inpainting>. Accessed: 2025-05-14. 2024.
- [21] Diffusers Team. *Stable Diffusion XL 1.0 Inpainting 0.1*. <https://huggingface.co/diffusers/stable-diffusion-xl-1.0-inpainting-0.1>. Accessed: 2025-05-14. 2023.
- [22] Wikipedia contributors. *Théâtre D'opéra Spatial – Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/wiki/Th%C3%A9%C3%A2tre\\_D%27op%C3%A9ra\\_Spatial](https://en.wikipedia.org/wiki/Th%C3%A9%C3%A2tre_D%27op%C3%A9ra_Spatial). Accessed: 2025-05-16. 2025.
- [23] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He и Jiajun Liang. «EAST: An Efficient and Accurate Scene Text Detector». B: *arXiv preprint arXiv:1704.03155* (2017). Accessed: 2025-05-14. URL: <https://arxiv.org/abs/1704.03155v2>.