Отчёт по итоговому проекту

Студент: Крамин Мурат Тимурович

Тема: Генерация уличных панорам на основе спутниковых и аэрофотоснимков

1. Введение

Описание проблемы

Современные геоинформационные системы, навигационные приложения и сервисы дополненной реальности нуждаются в детальном и актуальном визуальном описании городской среды. Панорамные изображения улиц позволяют решать задачи навигации, визуального планирования, кадастрового анализа и мониторинга инфраструктуры. Однако их получение требует затратных наземных съёмок, которые часто невозможны из-за сезонных, юридических или технических ограничений.

Синтез уличных панорам на основе доступных спутниковых или аэрофотоснимков представляет собой перспективное направление, позволяющее восполнять недостающие визуальные данные без необходимости физического присутствия. При этом требуется решение сложной задачи преобразования вида сверху в реалистичную панораму уровня улицы с учётом особенностей городской сцены.

Зачем это нужно

Разработанный подход актуален для:

- отечественных навигационных платформ и картографических сервисов;
- муниципальных служб, занимающихся планированием городской среды;
- систем дополненной и смешанной реальности;
- визуализации маршрутов в логистике и туризме.

Создание отечественного инструмента синтеза панорам позволит снизить зависимость от зарубежных решений и учесть специфику российских городов — как в архитектуре, так и в климате.

Постановка задачи

Цель проекта — разработать и обучить модель генерации уличных панорам на основе спутниковых снимков, адаптированную к условиям российской застройки. Для этого:

- сформирован датасет из более чем 18 000 пар «спутниковый снимок – панорама», покрывающий 76 крупнейших российских городов;
- автоматически сгенерированы семантические маски для каждой сцены;
- предложена архитектура генеративной модели с двойным выходом (панорама и сегментация) и обратной связью;
- реализован веб-интерфейс, позволяющий интерактивно выбрать координату и получить сгенерированную панораму в реальном времени.

Стек технологий

- **Язык программирования:** Python 3
- **Фреймворки:** PyTorch, torchvision, HuggingFace Transformers (SegFormer)
- **Библиотеки:** NumPy, OpenCV, PIL, pandas, scikit-learn, Streamlit, Folium
- **API и данные:** Google Street View API, спутниковые карты в проекции Web Mercator
- **Модель:** модифицированная архитектура U-Net с петлёй обратной связи, двумя дискриминаторами и мультизадачным обучением
- **Развёртывание:** веб-интерфейс на Streamlit с генерацией изображений по клику на карту

Этапы работы

1. **Сбор данных:** автоматическая выгрузка панорам, спутниковых снимков и генерация семантических масок.
2. **Предобработка и фильтрация:** валидация геометрии, нормализация, синхронизация трёх модальностей.
3. **Разработка архитектуры модели:** генератор с двойным выходом и итеративной корректировкой, каскад дискриминаторов.
4. **Обучение и оценка:** тренировка на GPU с визуализацией прогресса и подсчётом метрик SSIM, PSNR, IoU.
5. **Развёртывание:** создание веб-приложения, генерирующего панорамы по координатам пользователя.
6. **Анализ и выводы:** интерпретация результатов, выявление сильных и слабых сторон модели, перспективы развития.

2. Обзор существующих решений

Задача генерации уличных панорам по спутниковым изображениям активно исследуется в последние годы в связи с ростом интереса к автономной навигации, цифровому градостроительству и дополненной реальности. Ниже кратко изложены основные подходы, применяемые в современных системах.

Существующие подходы и алгоритмы

1. **Пиксель-ориентированные сети**
Ранние работы применяли сверточные сети для прямого преобразования спутникового вида в изображение улицы, обучаясь на парах изображений без дополнительной информации. Хотя такие модели просты в реализации, они часто приводят к размытым результатам и плохо справляются с редкими объектами.
2. **Модели с дополнительными каналами**
Введение структурных признаков — семантических масок, карт глубины или координат — улучшает устойчивость генерации. Такие модели требуют предварительной сегментации и усложняют пайплайн, но повышают качество при разнообразной застройке.
3. **Итеративные схемы с обратной связью**
Некоторые архитектуры реализуют повторную генерацию с учётом промежуточных признаков, что позволяет уточнять детали сцены. Эти решения демонстрируют лучшие метрики при сопоставимых вычислительных затратах.
4. **Усложнённые функции потерь**
Помимо стандартной GAN-потери, используются перцептивные расстояния, KL-дивергенция по распределениям классов и совместная оптимизация по RGB и семантическим слоям.
5. **Геометрические и параметрические модели**
Исследуются методы, учитывающие проекционную геометрию панорамы. Они обеспечивают точность размещения горизонта, но требуют внешних данных (например, LiDAR).
6. **Диффузионные и трансформерные модели**
Новейшие подходы используют диффузионные процессы с глобальным вниманием. Они показывают впечатляющие результаты, но предъявляют высокие требования к ресурсам и данным.

Выбор методов для данного проекта

Большинство перечисленных решений основаны на данных из США, Европы или Китая. Эти модели плохо обобщаются на российские города из-за отличий в архитектуре, материалах и климате. Поэтому в данном проекте:

- сформирован собственный корпус «спутник – панорама» для 76 российских городов;
- предложена архитектура с итеративным уточнением результата (feedback loop);
- модель обучается на две задачи одновременно: генерация изображения и сегментации;
- дискриминаторы оценивают как визуальное, так и семантическое качество.

Такой подход объединяет лучшие идеи существующих работ (цикловая генерация, двойной выход, семантический контроль) и адаптирован к российским условиям.

3. Подготовка данных

Описание выбранного датасета

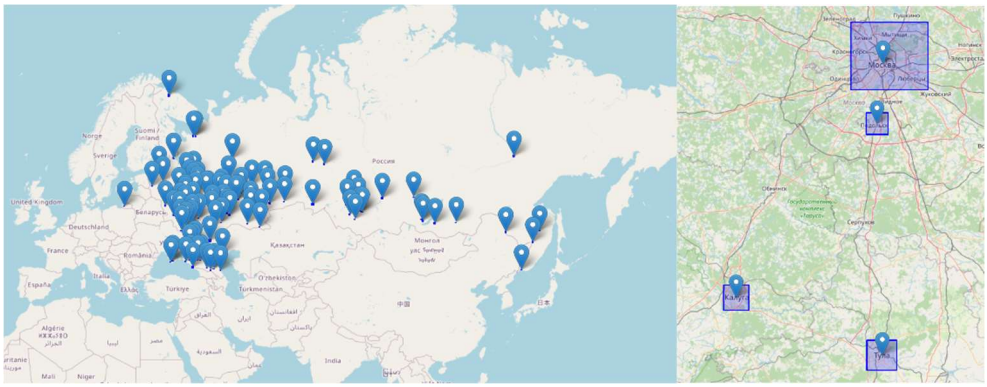

Рисунок 1 – География выборки: 76 крупнейших городов России

Для обучения модели был сформирован уникальный корпус данных, охватывающий 76 крупнейших городов России. В итоговый набор вошло **18 474 пары** изображений в формате «спутниковый снимок – уличная панорама», каждая из которых дополнена **семантической маской**, автоматически размеченной по 10 укрупнённым классам (небо, дорога, здания, деревья и др.). География выборки представлена на рисунке 1. Рисунок 1 демонстрирует общее распределение точек по территории России, включая регионы за пределами обучающего корпуса, а также области вокруг городов в столичном регионе.

- **Панорамы:** цилиндрические проекции с углом обзора 360° и вертикалью ~120°, разрешение не ниже 3584×1300 пикселей.
- **Спутниковые снимки:** квадратные фрагменты 256×256 пикселей, полученные в проекции Web Mercator на уровне детализации ~0.15 м/пикс.
- **Семантические маски:** индексированные изображения 256×1024, с классами, агрегированными из выходов модели SegFormer, дообученной на датасете ADE20K.

Предобработка данных

Весь цикл подготовки данных реализован автоматически:

- **Фильтрация:** отобраны только те сцены, где:
  - присутствует не менее 5% неба;
  - допустимая высота панорамы (1100–1400 пикселей);

- доля нераспознанных пикселей в маске не превышает 1%.
- **Нормализация:**
  - спутниковые изображения преобразованы к размеру 256×256;
  - для устойчивости к ориентации создаются четыре версии спутникового снимка с поворотом на 90° и конкатенацией по ширине (итоговый размер 256×1024).
  - панорамы и маски масштабированы до 256×1024 с сохранением пропорций.
- **Форматы хранения:** все изображения сохранены в сжатом виде, маски — как PNG и .npz (индексные массивы классов).

Визуализация ключевых аспектов

Для контроля качества разметки и корректности сопоставления «спутник – панорама – маска»:

Рисунок 2 – Примеры реальных панорам и соответствующих семантических масок с подписями классов объектов

- визуализировались примеры пар панорама–маска с наложением контуров и подписями классов (см. рисунок 3);
- строились гистограммы распределения классов в семантических масках;
- отображались карты ошибок по L1-показателю и различиям в сегментации.

В результате получен компактный и структурированный датасет, полностью адаптированный для задачи генерации уличных панорам и пригодный для масштабируемого обучения нейросетевой модели, охватывающего широкую географию, представленную на рисунке 1.

4. Разработка модели

Архитектура нейросети

В основе решения лежит генеративная архитектура типа **«генератор + двойной дискриминатор»**, ориентированная на одновременный синтез уличной панорамы и соответствующей ей семантической карты. Генератор реализован на модифицированной архитектуре U-Net, дополненной каскадом обратной связи (feedback loop), как показано на рисунках 3 и 4. Такая конструкция позволяет не только декодировать изображение по принципу энкодер-декодер, но и встраивать промежуточную коррекцию через блоки, получающие сигналы от дискриминаторов. Это усиливает способность модели к уточнению структурных элементов (границы зданий, линия горизонта, дороги и пр.).

Общая структура решения показана на рисунке 3. Модель принимает спутниковое изображение и итеративно генерирует два результата: RGB-панораму и соответствующую ей сегментационную карту. Используются два дискриминатора, контролирующих качество как визуального, так и семантического выхода.

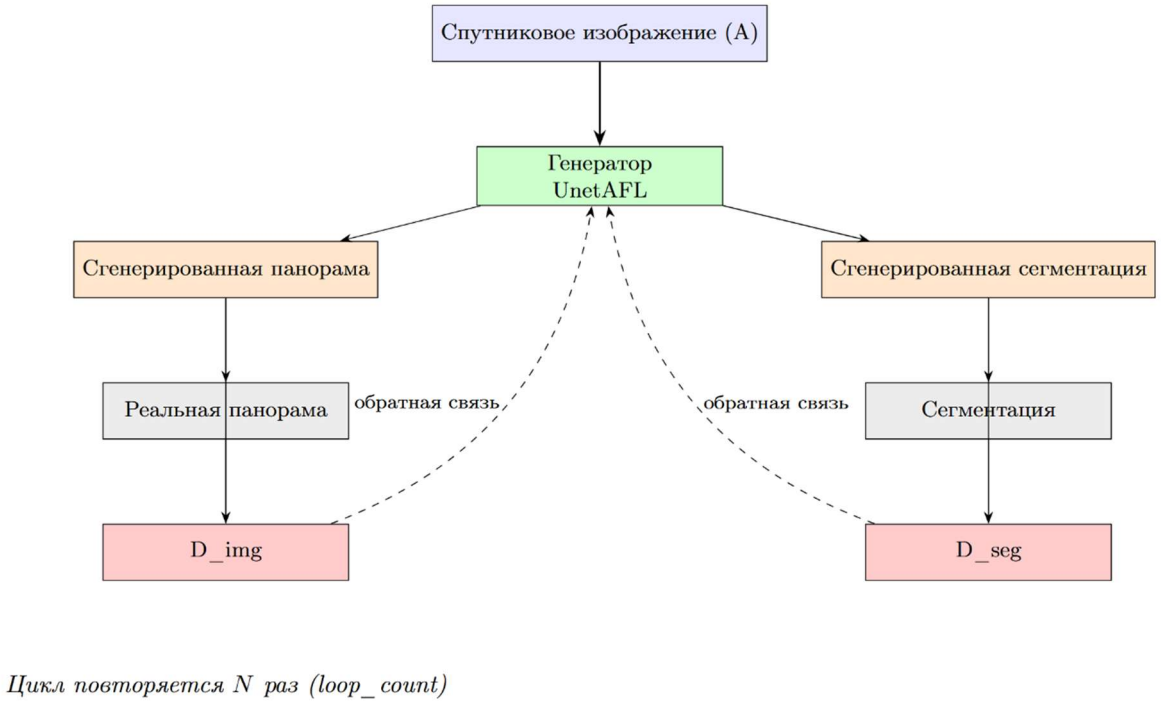

Рисунок 3 – Схема общей архитектуры генеративной модели с двойным выходом и обратной связью

Генератор состоит из:

- энкодера с четырьмя уровнями downsampling;
- блока остаточных преобразований (ResNet-блоки);
- декодера с симметричными skip-соединениями;

- трансформационных блоков, которые на каждом цикле уточнения получают признаки от дискриминаторов и возвращают их в генератор.

Выходной тензор разделяется на два канала:

- RGB-панорама;
- семантическая маска.

Такой подход обеспечивает одновременное обучение по визуальному и семантическому критерию, а наличие обратной связи позволяет постепенно уточнять мелкие детали сцены (например, фасады зданий и горизонт).

Каждый декодирующий уровень генератора дополнен собственным модулем обратной связи (feedback-блоком), который получает признаки от дискриминаторов и возвращает их в декодер. Эти блоки встроены в структуру модели таким образом, что итеративно уточняют декодируемые признаки, опираясь на визуальные и семантические ошибки предыдущего прохода. Такая организация позволяет добиться высокой структурной согласованности и коррекции локальных артефактов. Подробная схема взаимодействия компонентов представлена на рисунке 4.

Подробная структура генератора изображена на рисунке 4. Архитектура сочетает энкодер из сверточных блоков, промежуточную каскадную трансформацию на основе ResNet, и декодер с блоками апсемплинга и механизмами обратной связи. Каждый уровень декодера дополнительно уточняется соответствующим блоком Feedback, усиливая согласованность с дискриминатором.

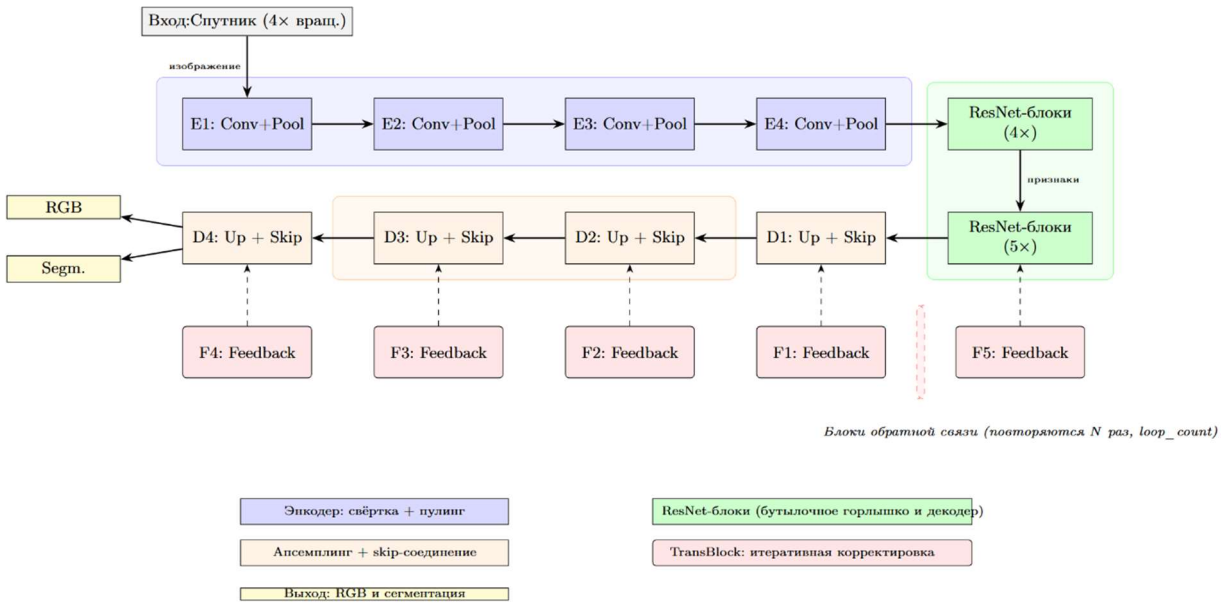

Рисунок 4 – Подробная архитектура генератора с каскадом обратной связи и мультизадачным декодером

Дискриминаторы (изображения и маски) построены по принципу **пирамиды признаков (Feature Pyramid Discriminator)**, позволяющей оценивать как локальные, так и глобальные структуры изображения.

Обоснование выбора метода

Выбранная архитектура сочетает в себе:

- проверенные компоненты (U-Net, PatchGAN-дискриминатор, L1 и GAN-потери);
- элементы современных исследований (feedback-обучение, мультизадачный выход);
- адаптацию под ограниченные вычислительные ресурсы (модель работает на одной видеокарте уровня RTX 3060).

В отличие от «одноразовой» генерации, петля обратной связи позволяет достигать более высокой структурной согласованности, что особенно важно при синтезе сложных городских сцен.

Гиперпараметры и стратегия обучения

- **Оптимизатор:** Adam ($\beta_1 = 0.5$, $\beta_2 = 0.999$)
- **Начальное значение learning rate:** $2 \cdot 10^{-4}$
- **План обучения:**
  - первые 15 эпох — стабильное обучение;
  - далее 30 эпох — линейное затухание learning rate.
- **Loss-функции:**
  - GAN-потери: для изображений и масок (классический логистический GAN);
  - L1-потери: между сгенерированной и эталонной панорамой и маской;
  - итоговая функция:

$$LG = 1k \sum i =$$

$$1k[LGANimg + LGANseg + \lambda img \cdot LL1img + \lambda seg \cdot LL1seg], \mathcal{L}_{\mathcal{G}}$$

$$= \frac{1}{k} \sum_{i=1}^{k} \left[\mathcal{L}_{\mathcal{GAN}}^{img} + \mathcal{L}_{\mathcal{GAN}}^{seg} + \lambda_{img} \cdot \mathcal{L}_{\mathcal{L}1}^{img} + \lambda_{seg} \cdot \mathcal{L}_{\mathcal{L}1}^{seg} \right],$$

где k=3k = 3 — число циклов обратной связи.

Подобранные значения параметров позволяют достичь баланса между визуальным качеством и устойчивостью обучения, при этом поддерживая разумное время инференса — ~0.8 изображений/сек на GPU.

5. Реализация и обучение

Описание кода и окружения

Проект реализован на языке Python с использованием фреймворка **PyTorch** для построения и обучения нейросетей. Вспомогательные задачи — визуализация, логгирование, загрузка данных и сохранение результатов — реализованы с использованием библиотек torchvision, NumPy, OpenCV, matplotlib, tqdm.

Обучение модели проводилось в среде с GPU NVIDIA RTX 3060 (12 GB), что обеспечило стабильную работу с батчами размером до 16 изображений. Логика генерации, дискриминации и обучения разделена по модулям, что позволило переиспользовать одни и те же компоненты в разных экспериментах (например, с отключённой сегментацией или без обратной связи).

Для удобства оценки результатов реализованы функции сохранения промежуточных изображений и графиков обучения, а также итоговых метрик.

Процесс обучения

- **Количество эпох:** 45 (15 — прогрев, 30 — с затуханием learning rate)
- **Лосс-функции:**
  - *L1 loss* между сгенерированной и реальной панорамой;
  - *L1 loss* между сгенерированной и эталонной семантической маской;
  - *GAN loss* (логистическая) — отдельно для RGB- и сегментных каналов;
  - Общий лосс составляется как сумма этих потерь с весами $\lambda img = 100 \lambda_{img} = 100, \lambda seg = 10 \lambda_{seg} = 10$.
- **Оптимизатор:** Adam ($\beta_1 = 0.5$, $\beta_2 = 0.999$)

Графики обучения

В процессе обучения автоматически сохранялись кривые:

- Значений функции потерь по эпохам (общая, по каждому из каналов отдельно);
- Визуализации синтезов после каждой эпохи (для визуальной оценки прогресса).

На рисунках 5–8 приведены ключевые графики, отражающие динамику обучения модели. Это позволяет наглядно оценить устойчивость и сбалансированность обучения между генератором и дискриминаторами, а также эффективность работы по отдельным каналам (RGB и сегментация).

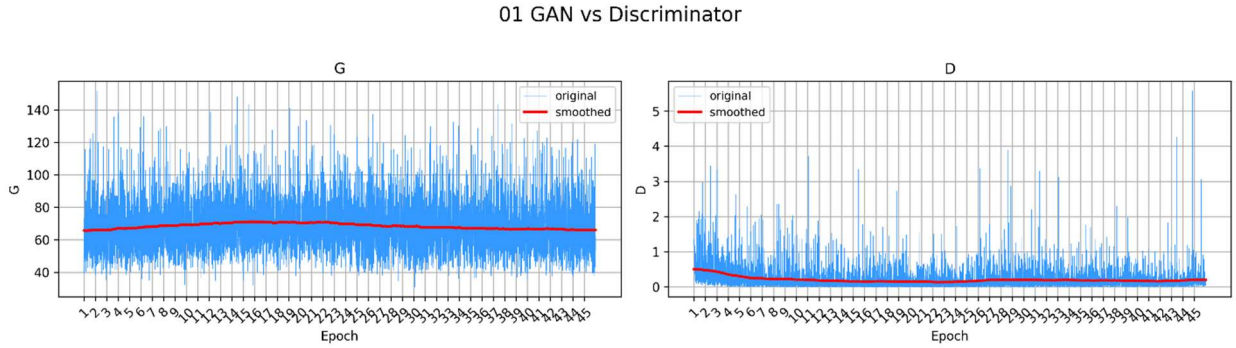

Рисунок 5 – Динамика общей GAN-функции генератора и дискриминатора по эпохам

Как видно на рисунке 5, значения функции потерь генератора (G) остаются в стабильном диапазоне, в то время как дискриминатор (D) быстро достигает устойчивости, свидетельствуя о сходимости процесса adversarial-обучения.

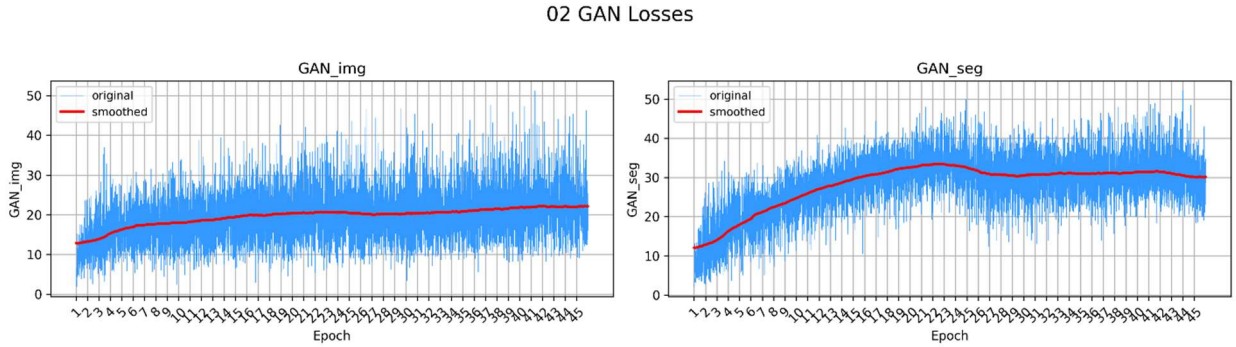

Рисунок 6 – Динамика потерь GAN по RGB и сегментационному каналам

Рисунок 6 демонстрирует, как распределяются GAN-потери по двум выходам: изображениям и сегментационным маскам. Обе кривые показывают рост в начале, затем стабилизацию — это подтверждает, что модель постепенно фокусируется на визуальных и семантических признаках.

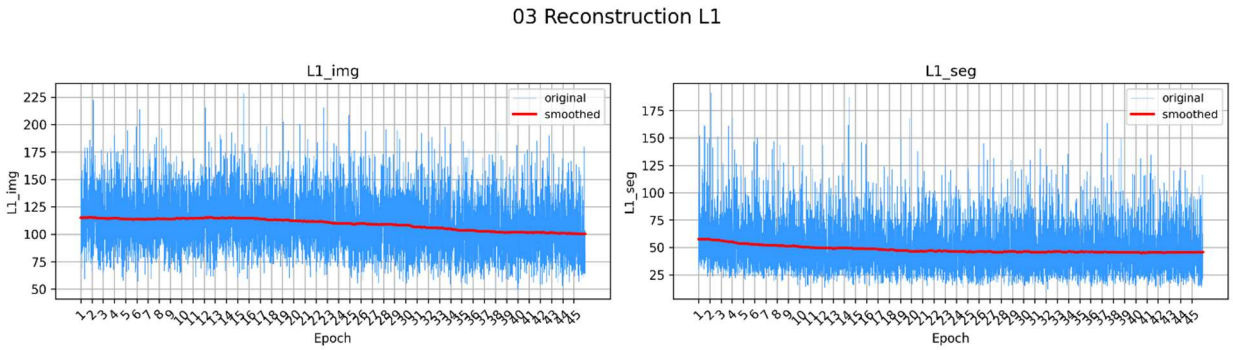

Рисунок 7 – Значения L1-потерь генерации изображений и сегментаций

На рисунке 7 представлены L1-потери. Видно, что средняя ошибка реконструкции по пикселям снижается в обоих каналах. Это говорит о постепенной коррекции мелких деталей панорамы и структуры сегментации.

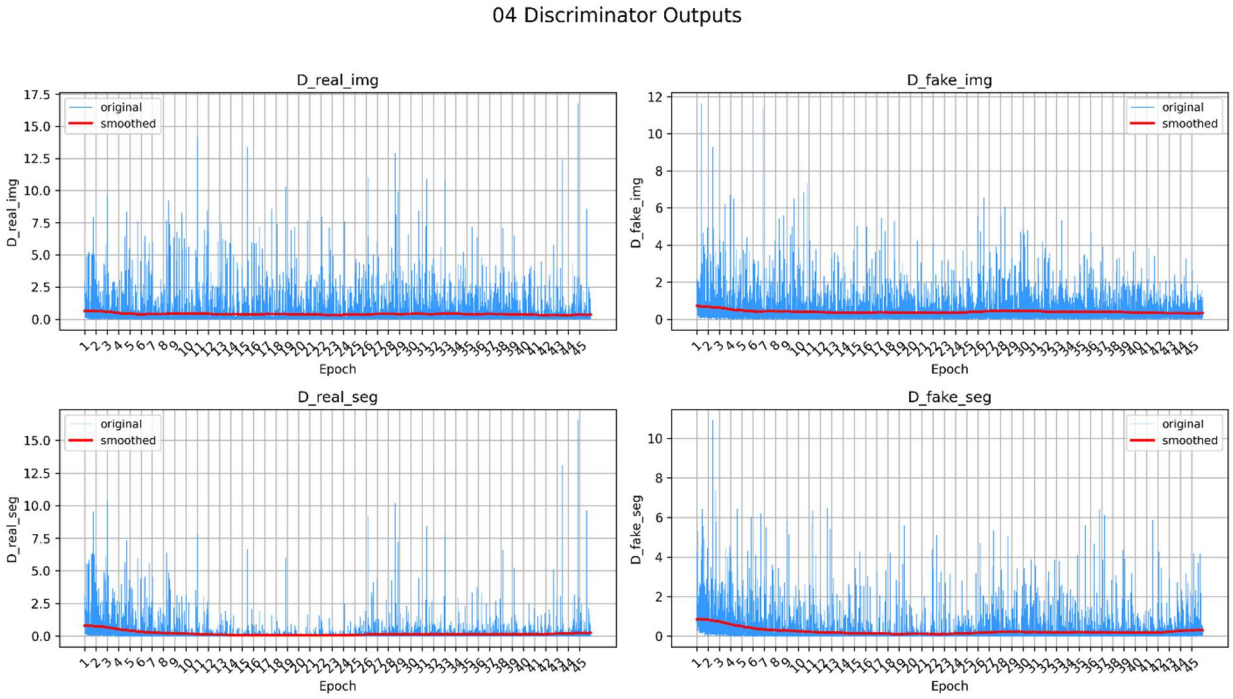

Рисунок 8 – Оценки дискриминаторов: для реальных и сгенерированных RGB и масок

Оценки дискриминаторов (рисунок 8) показывают уверенное различие между реальными и сгенерированными данными, особенно на ранних эпохах. При этом значения сглаживаются ближе к концу обучения, что соответствует равновесию между генерацией и дискриминацией.

Такой набор наблюдений позволил не только отслеживать устойчивость модели, но и своевременно выявлять переобучение или стагнацию.

6. Оценка качества модели

Метрики

Для оценки качества генерации были использованы следующие метрики:

- **SSIM (Structural Similarity Index):**

$$\text{SSIM} = 0{,}547$$

Значение выше 0.5 свидетельствует о сохранении более половины текстурных и структурных особенностей сцены. На практике это соответствует визуально различимым контурам зданий, деревьев и дорог.

- **PSNR (Peak Signal-to-Noise Ratio):**

$$PSNR = 15{,}8 \text{ д}$$

Это значение соответствует умеренному уровню искажений. На изображениях заметны отклонения в мелких деталях и области неба, но сохраняется общее соответствие сцене.

- **L1 Loss:**

Средняя абсолютная ошибка = 0,1106

Является базовым ориентиром расстояния между пикселями сгенерированного и настоящего изображения.

- **KL-дивергенция по классам сегментации:**
  - Средняя: 1.58 ± 0.71
  - Top-1 класс: 0.0001
  - Top-5 классов: 0.417

  Эти значения показывают, насколько модель сохраняет статистику распределения объектов сцены, особенно важных для городской навигации (здания, дорога, небо).
- **Показатели сегментации:**
  - Pixel Accuracy: **0.9659**
  - Mean IoU: **0.2445**
  - Class-wise IoU (по 10 классам): от 0.0 до 0.966

  Высокая точность достигнута по фоновым классам (небо, дорога), однако редкие классы (машины, люди, столбы) были слабо представлены и практически не реконструируются.

Confusion matrix и интерпретация ошибок

Для семантической маски построена матрица ошибок между предсказанными и реальными классами. Она показала:

- сильную путаницу между схожими по текстуре классами (например, здания и заборы);
- тенденцию к «затиранию» редких объектов — они часто заменяются фоновыми участками;
- хорошее соответствие по основным классам ландшафта, что важно для общей структуры сцены.

Тестирование на новых данных

Модель была протестирована на отдельной выборке городов, не входивших в тренировочный набор (например, Тула, Калуга, Пермь). Качественные результаты сохранились:

- не искажены пропорции зданий;
- восстановлены горизонт и перспектива;
- сохранены ключевые классы в сегментации (небо, дорога, фасады).

Для демонстрации генерализации модели были проведены визуальные тесты на ранее не встречавшихся сценах. На рисунках 9 и 10 показаны результаты работы генератора при различных ориентациях входного спутникового изображения. Видно, что модель сохраняет общую структуру сцены (дороги, небо, границы деревьев и зданий), несмотря на отсутствие данных из этих локаций в тренировочном наборе.

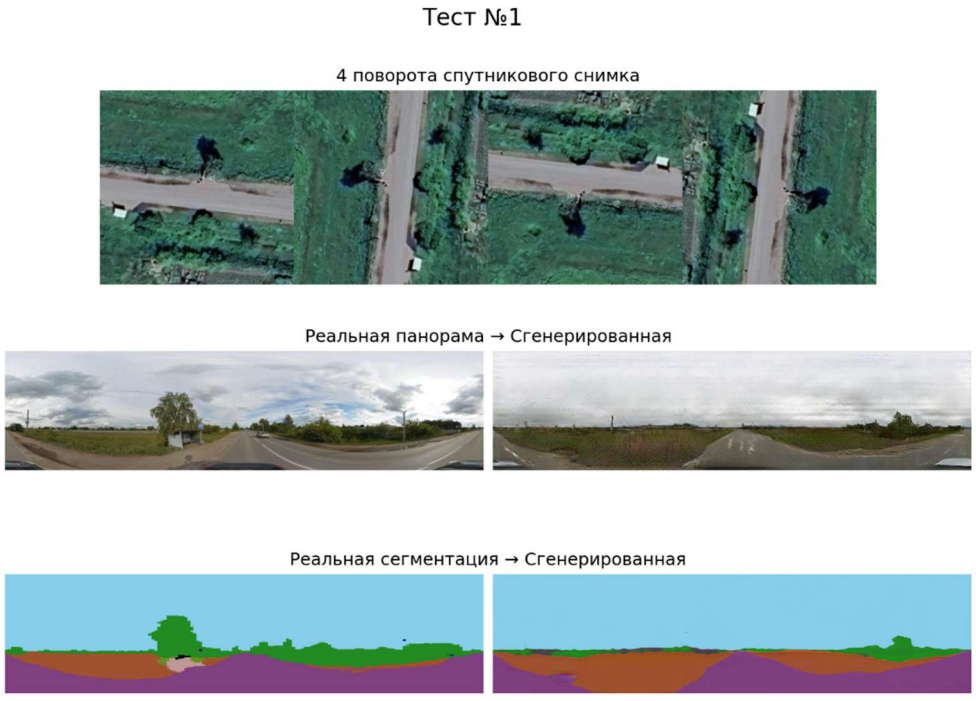

Рисунок 9 – Результаты генерации панорамы и сегментации на новой сцене (Тест №1)

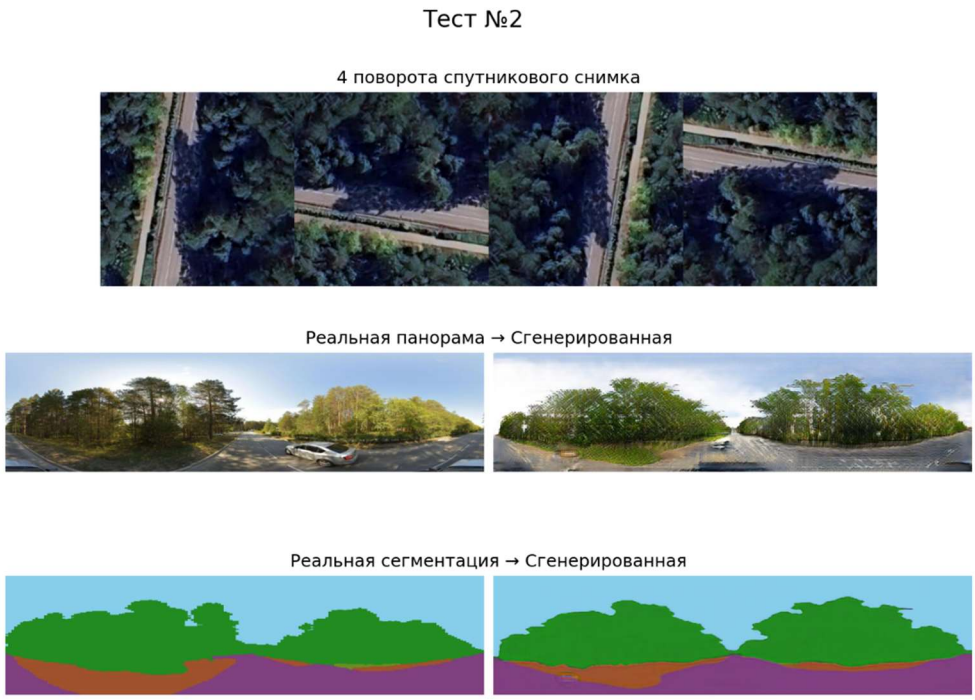

Рисунок 10 – Результаты генерации панорамы и сегментации на новой сцене (Тест №2)

Отдельно стоит отметить, что модель стабильно реконструирует форму горизонта и крупные ландшафтные элементы даже при наличии сильной текстурной неоднородности (см. Тест №2). При этом текстуры мелких объектов в некоторых случаях упрощаются, что соответствует ранее выявленным ограничениям архитектуры.

Таким образом, итоговая модель демонстрирует устойчивость к географической вариативности и генерализуемость на незнакомых регионах.

7. Развертывание и демонстрация

Использование модели в продакшене

Для демонстрации прикладной ценности модели был реализован полноценный прототип веб-приложения, позволяющий в интерактивном режиме синтезировать уличную панораму по спутниковому изображению. Пользователь может выбрать любую точку на карте, после чего автоматически загружается соответствующий спутниковый фрагмент, подаётся в генератор, и в течение нескольких секунд отображается сгенерированная панорама и семантическая маска.

Такой интерфейс может служить основой для:

- интеграции в геосервисы;
- визуального планирования городской среды;
- задач в дополненной реальности;
- автономных систем навигации в новых районах.

Разработка веб-интерфейса

Интерфейс разработан с использованием библиотеки Streamlit и интеграции Folium для отображения интерактивной карты с возможностью выбора произвольной координаты (рисунок 11). После клика загружается соответствующий спутниковый фрагмент, который автоматически преобразуется в уличную панораму. Это обеспечивает интуитивный и наглядный пользовательский опыт. Основной функционал включает:

- карту с возможностью выбора точки кликом;
- кнопку генерации, инициирующую запуск модели;
- отображение результата (панорама и сегментация);
- сохранение полученного изображения;
- статус выполнения и защита от повторного запуска.

Для демонстрации работоспособности модели в прикладной среде был создан удобный веб-интерфейс, позволяющий пользователю выбрать точку на карте и в реальном времени получить сгенерированную панораму. Рисунок 11 иллюстрирует процесс: после клика по координатам загружается спутниковый снимок и отображается результат работы генератора.

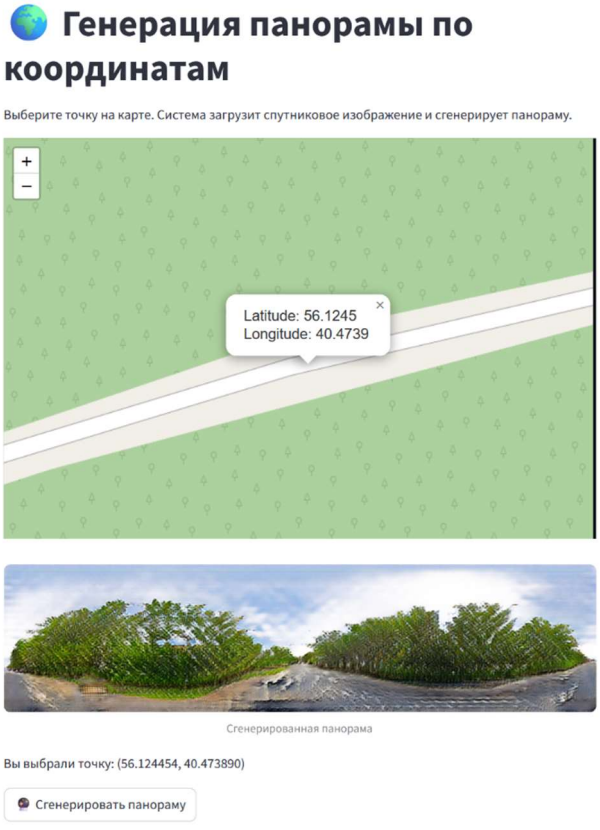

Рисунок 11 – Веб-интерфейс генерации панорамы по координатам в Streamlit

Такой визуальный формат может быть легко интегрирован в существующие геосервисы, муниципальные интерфейсы планирования городской среды или приложения в области логистики, туризма и дополненной реальности. Наличие полноценной карты и мгновенного результата делает систему применимой даже без технической подготовки пользователя.

Архитектура устроена таким образом, что генерация происходит локально, с загрузкой предварительно обученной модели. Приложение поддерживает масштабирование и перенос на сервер.

Демонстрация

- Время генерации одной сцены — менее **2 секунд** на GPU и ~**10 секунд** на CPU.
- Интерфейс доступен в виде локального скрипта web_app.py, запускаемого одной командой.
- Поведение модели протестировано на различных координатах по территории России, включая города, не использованные в обучении.

Результаты демонстрируют практическую применимость подхода и возможность его интеграции в реальные геоинформационные и картографические системы.

8. Выводы и перспективы

Основные результаты работы

В ходе проекта была разработана и обучена нейросетевая модель, способная генерировать уличные панорамы по спутниковым снимкам с учётом российской архитектурной специфики. Построен полный пайплайн от сбора и предобработки данных до визуализации результата в веб-интерфейсе. Итоговая модель демонстрирует стабильное качество генерации:

- достигнуто значение SSIM = 0.547 и PSNR = 15.8 дБ,
- обеспечена корректная реконструкция ключевых классов сцены в семантических масках,
- разработан удобный пользовательский интерфейс для генерации изображений по координатам.

Проект показал, что возможно создавать реалистичные уличные виды в условиях отсутствия панорамных данных, опираясь только на доступные спутниковые изображения.

Нерешённые проблемы

Несмотря на достигнутые результаты, модель испытывает сложности в следующих аспектах:

- недостаточное качество генерации мелких объектов (транспорт, люди, вывески);
- размытые или схематичные текстуры в центральной части панорамы;
- слабая реконструкция динамичных или редко встречающихся элементов.

Также наблюдается неустойчивость при обработке регионов с малоэтажной и частной застройкой, где геометрическая структура сцены слабо выражена.

Возможности улучшения

В качестве направлений дальнейшей работы можно выделить:

- использование **диффузионных моделей** или **Vision Transformer** в генераторе;
- дообучение сегментатора на панорамах российских городов для улучшения масок;
- введение **глубинных карт** как дополнительного входа;
- реализацию **циклических или стилевых ограничений** (cycle-consistency, perceptual loss);
- генерацию не только панорам, но и **пошаговых видов**, пригодных для навигации;
- развертывание модели на сервере с API и возможностью пакетной обработки координат.

Таким образом, разработанная система может служить основой для дальнейших исследований в области визуального моделирования городской среды и найдет применение в ряде практических задач.