Генеративные модели в создании трехмерных объектов

1. **Работа с данными:**

Обработка 2D изображений - я рассматривал изображения были рассмотрены как одноканальные, поскольку так как анализ показал, что основное различие между пикселями заключается в контрастности, а не в цвете.

Для увеличения разнообразия тренировочной выборки и предотвращения переобучения применялась аугментация:

- Изменение яркости и контраста: для имитации различных условий освещенности.
- Аффинные преобразования: для случайных небольших поворотов и сдвигов.
- Горизонтальные отражения
- Гауссово размытие: для добавления небольшого уровня шума/размытия.

Работа с 3D объектами - Для работы с объектами я использовал библиотеку open3D. Объекты загружались в виде треугольных мэшей (набор вершин и графов), центрировались и масштабировались для того, чтобы все объекты были одинаково ориентированы и заняли один и тот же объем в пространстве.

2. **BaseLine**

В качестве базового решения была реализована архитектура, состоящая из:

- 2D-энкодера на основе предобученного ResNet 18 на ImageNet
- 3D-декодера с полносвязными слоями
- Loss-функция Chamfer Distance

Обоснование выбора ResNet 18 (см. рисунок 1):

1. Проверенная архитектура с хорошими трансферными свойствами

2. Баланс между скоростью и точностью по сравнению с иными моделями (U-Net, EfficientNet-B7)

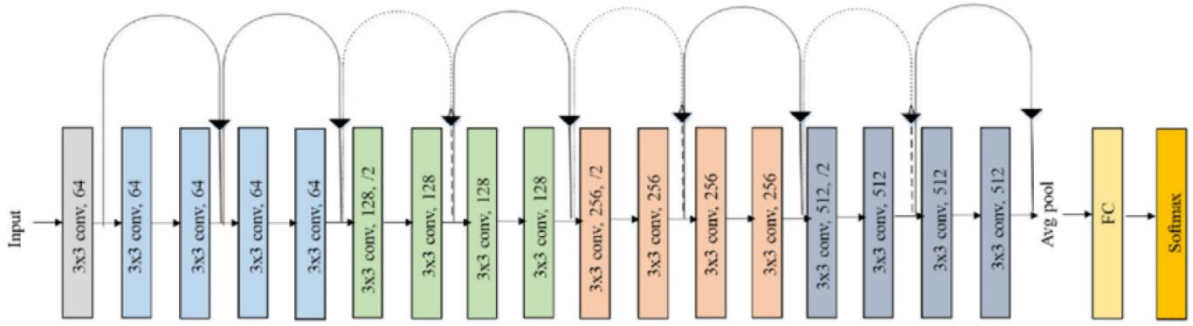

Рисунок 1. Архитектура ResNet 18

В декодере была реализована трехслойная MLP-структура, в качестве простой архитектуры для базового решения. Результатом работы декодера было облако точек (2048 точек). Базовое решение показало, что модель не могла грамотно отделить фон от объекта, а также дать глубину объекту (см. рисунок 2)

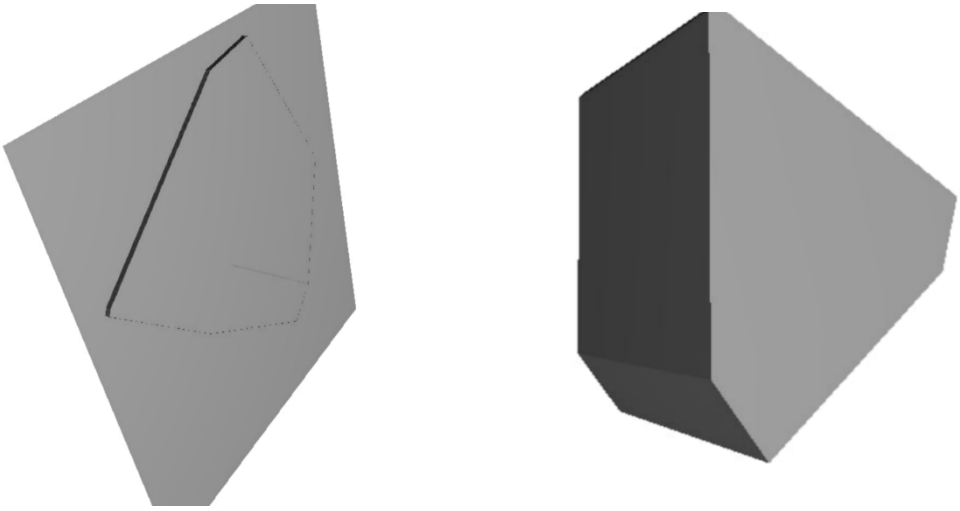

Рисунок 2. Слева - прогноз модели, справа - исходная деталь

Основная сложность, с которой столкнулся позже - дисбаланс классов. Количество пикселей фона превосходило количество пикселей модели (доля положительных пикселей - 2,5%). Chamfer Distance неустойчив к выбросам, если в одном из множеств точек присутствует хотя бы один значительный

выброс, даже единичный, то минимальное расстояние от этой аномальной точки до ближайшей нормальной точки из другого множества точек существенно увеличится. Это увеличение приведет к росту общего значения Chamfer Distance, искажая оценку качества совпадения форм. Дисбаланс классов был учтен при дальнейшем работе с моделями и выбором Loss-функций.

3. Полученные результаты на BaseLine:

   Train Loss: 0.0218, Val Loss: 0.0249

3. Research

Было проведено исследование, в ходе которого было изучено несколько современных подходов к 3D object reconstruction from a single image.

Изучение таких работ, как TripoSR, который успешно применяет трансформерную архитектуру для быстрой генерации 3D мешей, и Pix2Vox++, улучшающий воксельную реконструкцию с помощью многомасштабных признаков и внимания, выявило несколько ключевых закономерностей и эффективных архитектурных решений.

Было отмечено следующее:

- Доминирование Трансформеров в Качестве Энкодеров:

Многие современные модели для задач, требующих глубокого понимания содержимого изображения (включая оценку 3D геометрии из 2D данных), активно используют архитектуры на основе трансформеров (например, Vision Transformer, Swin Transformer). Их способность улавливать глобальные зависимости и контекст в изображении благодаря механизмам self-attention делает их мощным инструментом для извлечения признаков.

- Важность skip-connections в Архитектуре:

В задачах генерации или реконструкции, где требуется восстановить детализированный вывод из сжатого представления (как в нашем случае — 3D воксели из 2D признаков), архитектуры типа U-Net с использованием skip-connections являются стандартом. Skip-connections позволяют передавать

низкоуровневые, детализированные признаки из ранних слоев энкодера напрямую в соответствующие по разрешению слои декодера. Это помогает бороться с потерей информации энкодера и значительно улучшает реконструкцию мелких деталей и четкость границ.

Также были изучены неявные представления (NeRF, DeepSDF), но для генерации вокселей был выбран более прямой подход.

4. Решение

Финальная архитектура модели представляет собой энкодер-декодерную сеть:

- Энкодер (EnhancedEncoder):

Backbone: в качестве основы для извлечения признаков из входного 2D изображения был выбран предобученный Swin Transformer V2 tiny (см. рисунок 3). Этот выбор обусловлен способностью трансформеров эффективно улавливать как локальные, так и глобальные контекстные зависимости в изображении, а также благодаря проведенному исследованию.

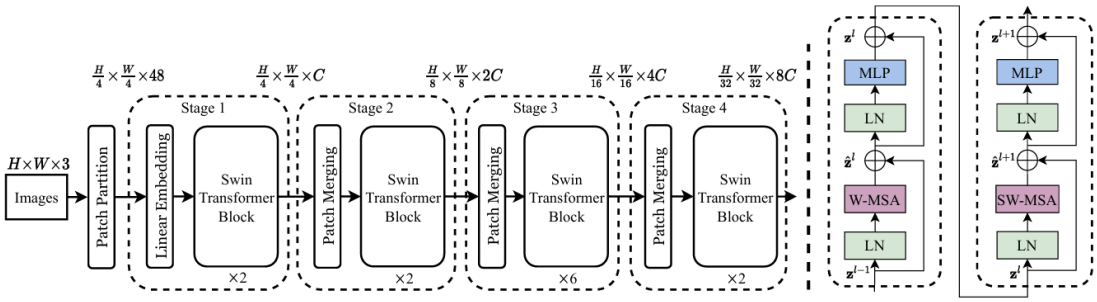

Рисунок 3. Архитектура Swin Transformer V2

- Извлечение признаков:

Энкодер настроен на извлечение иерархических карт признаков с нескольких стадий. Последний, самый глубокий уровень признаков, используется как основной вход для декодера, а предыдущие уровни предназначены для skip-соединений.

- Декодер (VoxelDecoder):

Формировался начального 3D объем, вместо полного сжатия 2D признаков в вектор, так как это могло привести к потере деталей. Используется

начальный 2D сверточный слой для адаптации признаков. Затем, с помощью операции изменения формы эти 2D признаки разворачиваются в низкоразрешающий 3D тензор, сохраняя пространственную информацию.

Последовательность из нескольких простых блоков, каждый из которых включает транспонированную 3D свертку, нормализацию по батчу и активацию ReLU, используется для постепенного увеличения пространственного разрешения 3D тензора до целевого размера воксельной сетки.

Заключительный 3D сверточный слой с ядром 1x1x1 преобразует многоканальный тензор в одноканальный выходной тензор логитов.

5. Результаты

Примечание: в качестве основной фигуры для визуализации работы модели был выбран цилиндр. Все дальнейшие изображения объектов будут показаны на фоне построения этого цилиндра (см. рисунок 4).

Рисунок 4. Тестовый цилиндр

Ключевым аспектом исследования был подбор эффективной функции потерь, способной справиться с сильным дисбалансом классов (объектные

воксели составляют ~2.57% от общего числа) и корректно направлять обучение модели.

1. Начальные Эксперименты:

Первоначальные попытки с использованием стандартного BCEWithLogitsLoss приводили к тому, что модель быстро обучалась предсказывать преимущественно фон, что давало низкий лосс, и высокий IoU (см. рисунок 5).

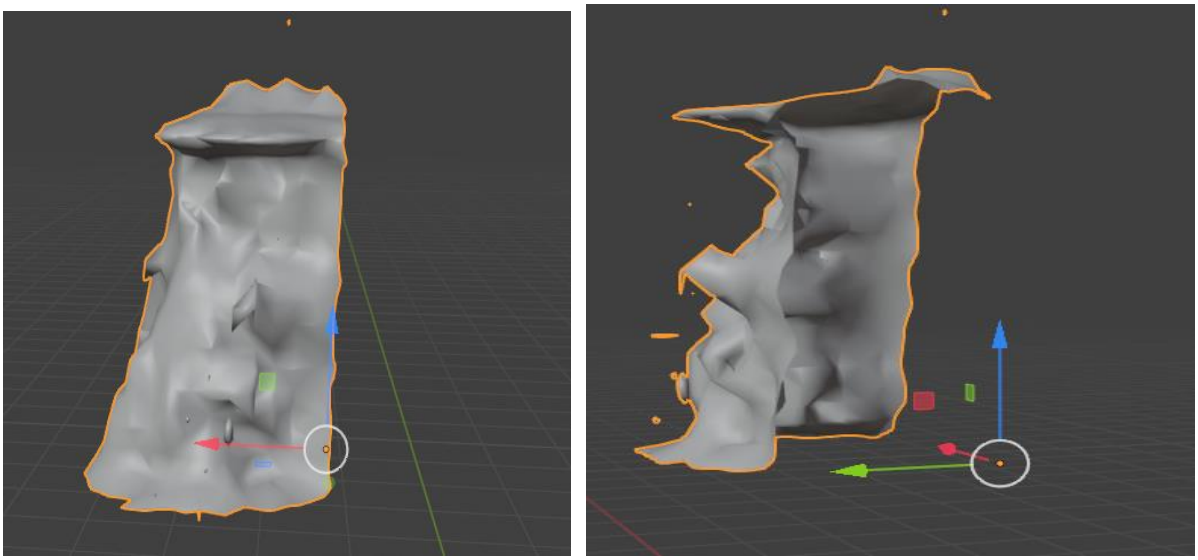

Рисунок 5. Предсказания модели с BCEWithLogitsLoss

2. Внедрение pos_weight:

Для борьбы с дисбалансом в BCEWithLogitsLoss был введен параметр pos_weight, рассчитанный как (1 - positive_ratio) / positive_ratio (около 37.85). Это дало значительное улучшение, и модель начала формировать правильную форму объекта (см. рисунок 6).

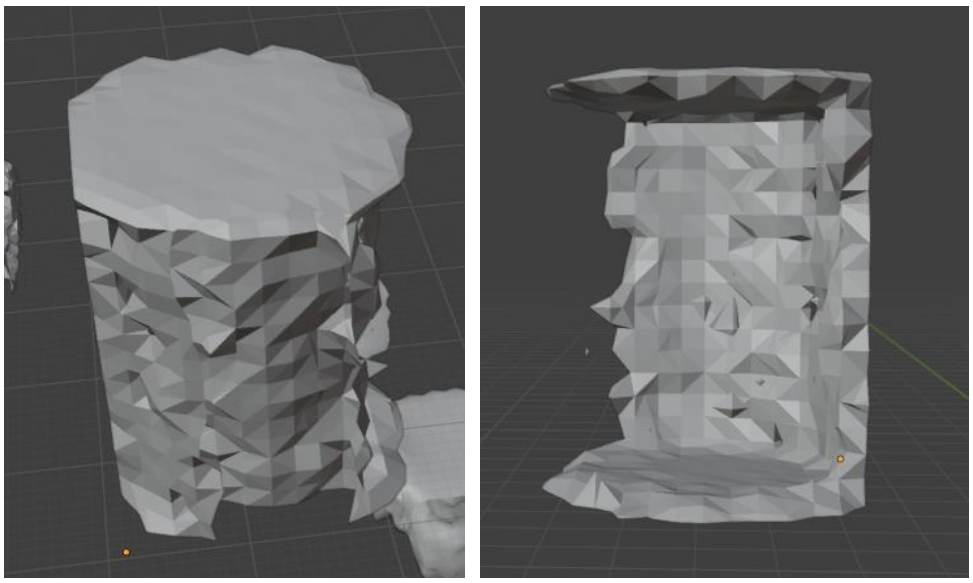

Рисунок 6. Предсказания модели с BCEWithLogitsLoss + pos_weight

Dice Loss и BCEWithLogitsLoss (ComboLoss): Для дальнейшего улучшения качества сегментации и формы был внедрен Dice Loss, который устойчив к дисбалансу классов и акцентирует внимание на мелких деталях. Это помогло сформировать полноценный цилиндр (см. рисунок 7):

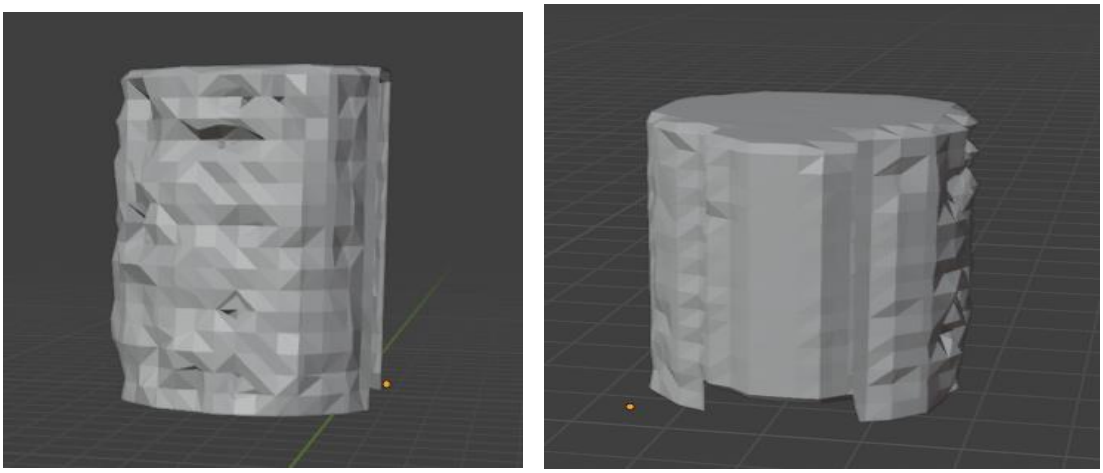

Рисунок 7. Предсказания модели с BCEWithLogitsLoss + Dice Loss

Также был опробован Focal Loss + Twersky loss. Это привело к сглаженным формам цилиндра, но сама форма объекта стала менее похожа на исходное изображение (см. рисунок 8).

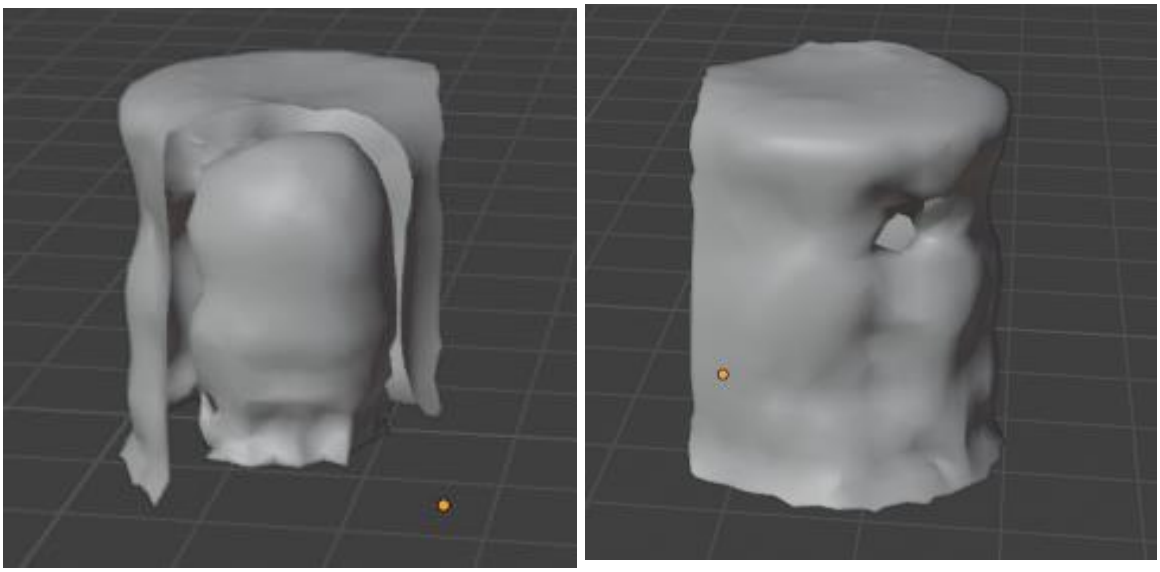

Рисунок 8. Предсказания модели с Focal Loss + Twersky loss

Оптимизатор и Планировщик:

- Использовался AdamW с weight_decay=0.01. Learning rate (изначальный 1e-3) подбирался и корректировался с помощью ReduceLROnPlateau, отслеживающего Val Loss (mode='min').

Обучение:

- Модель обучалась на 50 эпохах с использованием Early Stopping для предотвращения переобучения (с patience 15 эпох) и сохранением весов модели, показавшей лучший Val IoU.

Параметризация модели:

Текущая архитектура генерирует воксельное представление объекта. Для перехода к выводу параметрических моделей, таких как набор числовых параметров (длина, ширина, радиус) или последовательность CAD-команд (аналогично подходу DeepCAD), нужно изменить вход модели (облако точек) и обучающие данные. Необходимы датасеты, где 2D изображения

сопоставлены с точными списками параметров сборки объектов в CAD-системах.

Архитектура модели также потребует значительной переработки. Энкодер, извлекающий признаки из изображения (Swin Transformer), вероятно, может быть адаптирован. Однако декодер придется проектировать заново: вместо построения воксельной сетки, он должен будет генерировать либо вектор числовых значений, либо последовательность текстовых токенов (CAD-инструкций).

Одним из перспективных, по моему мнению, направлений мог бы стать подход DeepCAD, где для генерации CAD-инструкций и, следовательно, построения параметрического объекта, могут использоваться LLM или другие модели, ориентированные на работу с последовательностями. Плюсы такого подхода

- Полная совместимость с CAD-системами
- Возможность тонкого редактирования моделей
- Интерпретируемость результатов

Выводы

Лучший результат был получен с помощью BCEWithLogitsLoss + Dice Loss.

Ключевая проблема, с которой я столкнулся – переобучение.

Удалось добиться объекта цилиндрической формы. Также были рассмотрены и другие объекты, которая предсказала модель, в каждом объекте были свои проблемы в вокселизации, что говорит о необходимости провести анализ ошибок модели и изменить архитектуру / аугментацию / настройки модели.

Лучший результат (см. рисунок 9)

Train: Loss 0.3391 | IoU 0.6253

Val: Loss 0.3678 | IoU 0.6218

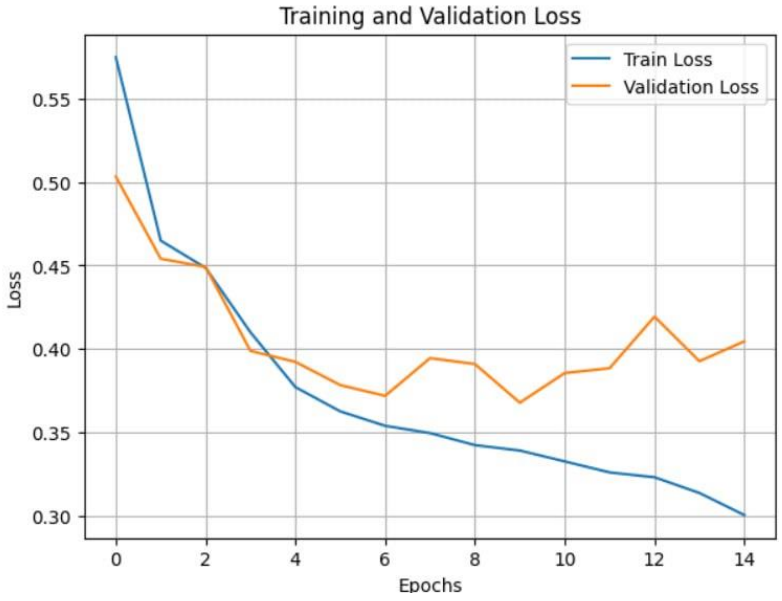

Рисунок 9. График Training and Validation Loss