Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»
Институт радиоэлектроники и информационных технологий - РТФ
Кафедра информационных технологий и систем управления

ДОПУСТИТЬ К ЗАЩИТЕ ПЕРЕД ГЭК

Зав. кафедрой ИТиСУ
Е.В. Кислицын
(подпись) (Ф.И.О.)
«03» 06 2024 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ИССЛЕДОВАНИЕ МЕТОДОВ СЕМАНТИЧЕСКОЙ СЕГМЕНТАЦИИ ДЛЯ ОБЪЕКТОВ ТИПА ПРОЖИЛКИ

Научный руководитель: Борисов Василий Ильич
к.т.н., доцент

подпись

Нормоконтролер: Бредихина Наталья Сергеевна

подпись

Студент группы: РИМ-220906
Мельников Владислав Андреевич

подпись

Екатеринбург
2024

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
**«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»**

Институт радиоэлектроники и информационных технологий - РТФ
Кафедра информационных технологий и систем управления
Направление подготовки 09.04.01 Информатика и вычислительная техника
Образовательная программа Инженерия искусственного интеллекта

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студента Мельникова Владислава Андреевича группы РИМ-220906

(фамилия, имя, отчество)

1. Тема выпускной квалификационной работы Исследование методов семантической сегментации для объектов типа прожилки

Утверждена распоряжением по институту от «4» декабря 2023 г. № 33.02-05/298

2. Научный руководитель Борисов Василий Ильич, доцент, кандидат технических наук

(Ф.И.О., должность, ученая степень, ученое звание)

3. Исходные данные к работе Датасет с камнями в открытом карьере

4. Перечень демонстрационных материалов *презентация в MS PowerPoint*

5. Календарный план

п/п	Наименование этапов выполнения работы	Срок выполнения этапов работы	Отметка о выполнении
1.	Глава 1. Анализ предметной области в сфере семантической сегментации	до 23.03.2024 г.	
2.	Глава 2. Исследование набора данных и анализ метрик	до 29.04.2024 г.	
3.	Глава 3. Тестирование нейросетевых алгоритмов	до 19.05.2024 г.	
4.	ВКР в целом	до 20.05.2024 г.	

Научный руководитель _____ Ф.И.О. _____ (подпись)

Студент задание принял к исполнению 12.03.2024 (дата) _____ (подпись)

6. Допустить Мельникова Владислава Андреевича к защите выпускной квалификационной работы в экзаменационной комиссии

Зав. кафедрой ИТиСУ _____ (подпись) Е.В. Кислицын Ф.И.О.

РЕФЕРАТ

Выпускная квалификационная работа содержит 55 страницы, 37 рисунков, 3 таблицы, 45 литературных источников.

Ключевые слова: СЕМАНТИЧЕСКАЯ СЕГМЕНТАЦИЯ, · СЕГМЕНТАЦИЯ ЭКЗЕМПЛЯРОВ, СВЁРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ, ОБНАРУЖЕНИЕ ОБЪЕКТОВ, АЛГОРИТМЫ ФИЛЬТРАЦИИ, ПРОБЛЕМА ГОРНОДОБЫВАЮЩЕЙ ПРОМЫШЛЕННОСТИ.

Объектом исследования являются цифровые изображения камней в открытом карьере.

Целью работы является разработка и реализация алгоритма детектирования и сегментации асбестовых прожилок с применением аппарата искусственного интеллекта.

В исследовании представлен аналитический обзор методов и существующих технических и программных систем, использующих методы искусственного интеллекта для сегментации на основных тестовых датасетах. Проведён анализ существующих моделей, протестированы новые модели на основе сверточных сетей (UNet и Attention Unet) и трансформеров (SegFormer), предложен лучший алгоритм для задачи сегментации асбестовых прожилок.

В результате применения модели искусственного интеллекта удалось эффективно решить задачу сегментации прожилок и достигнуть приемлемой точности полученных результатов при небольшой вычислительной мощности.

Областью применения разработанного алгоритма является не только его использование в рамках анализа содержания асбеста в снимках карьера. Полученные модели могут использоваться для определения дефектов на различной продукции и в медицине.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящей пояснительной записке к ВКР применяют следующие сокращения и обозначения:

МВ – attention mechanism

МСВ – multi head selfattention

СВ - selfattention

НС – нейронная сеть

BCE – Binary Cross Entropy

TP – True Positive

TN – True Negative

FP – False Positive

FN – False Negative

CNN – Convolutional neural network

ViT – Visual transformer

Swin Transformer – Shifted windows transformer

ВВЕДЕНИЕ

Асбест — это волокнистый силикатный минерал, который широко используется в строительных материалах благодаря своим полезным свойствам, таким как высокая предельная прочность на разрыв, низкая теплопроводность и относительная устойчивость к химическим воздействиям. Поскольку асбест состоит из микроскопических пучков силикатных волокон, асбестовые волокна могут переноситься по воздуху при механическом повреждении асбестосодержащих материалов или их разрушении в результате длительного воздействия солнечного света. Вдыхание асбестовых волокон повреждает легкие, что приводит к серьезным проблемам со здоровьем, таким как плевральная мезотелиома и рак легких. Заболевания, связанные с асбестом, ежегодно приводят к смерти примерно 255 000 человек во всем мире, а число случаев рака, связанного с асбестом, продолжает расти. Хотя использование асбеста в настоящее время запрещено во многих развитых странах, асбестосодержащие материалы по-прежнему остаются в старых зданиях, что приводит к образованию асбестовых волокон в воздухе. В США асбест стал причиной более 200 000 смертей за последние несколько десятилетий; даже сегодня ежегодно диагностируется около 2000-3000 новых случаев мезотелиомы, рака, связанного с асбестом [1]. Поэтому принятие мер по удалению асбеста из горных материалов имеет решающее значение для предотвращения его воздействия.

Актуальность работы. Основная проблема современной добычи заключается в том, что геологи зачастую как правило не могут описать критерии оценки содержания асбеста в камнях. Зачастую эти критерии объясняются лишь наличием опыта и собственноручных расчетов. Такие методы оценки трудно описать с научной точки зрения. К тому же обучение таких специалистов дорогостоящие и занимает много времени. На сегодняшний момент лабораторная оценка требует высоких временных затрат и специальное оборудование для определения асбестовых прожилок. Все

вышеперечисленные проблемы заставляют задуматься над созданием автоматизированной системы, на основе моделей компьютерного зрения, которая поможет при добыче полезных ископаемых открытым способом.

Объект исследования – системы компьютерного зрения для оценки выхода продукции.

Предмет исследования – архитектуры нейронных сетей для задач сегментации продукции на изображениях.

Цель работы. Целью работы является разработка модели для быстрой и точной оценки процентного содержания асбеста в ископаемых, в условиях открытого карьера с применением методов компьютерного зрения.

Постановка задачи. Достижение поставленной цели предполагает решение следующих задач:

1. Аналитический обзор существующих нейросетевых алгоритмов семантической сегментации;
2. Исследование набора данных и постановка экспериментов
3. Тестирование и сравнение нейросетевых алгоритмов

1. Обзор алгоритмов и инструментов для задач семантической сегментации

1.1. Семантическая сегментация

Семантическая сегментация — это задача компьютерного зрения, в которой изображение разбивается на сегменты и каждый пиксель относится к определенному классу или категории. В отличие от обычной классификации изображений, где целью является присвоение всему изображению одной метки класса, семантическая сегментация предоставляет детальную информацию о структуре объектов на изображении [9]. Процесс перехода от классификации к сегментации представлен на рисунке 1.

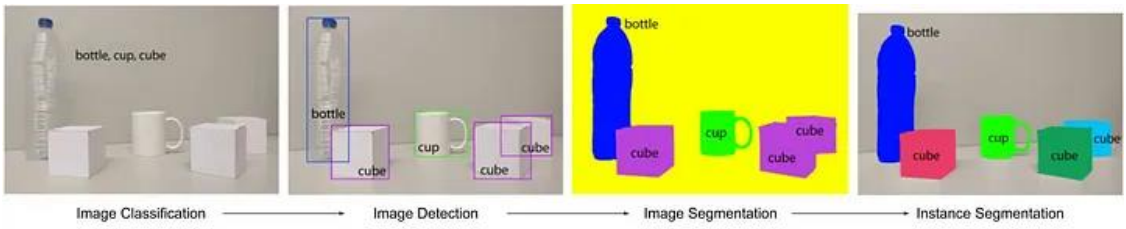

Рисунок. 1 - Иллюстрация перехода от грубых к тонким выводам

В настоящее время, существует множество нейросетевых методов семантической сегментации, которые могут быть классифицированы на две основные категории: методы на основе пикселей и методы на основе регионов.

Методы на основе пикселей, такие как Fully Convolutional Networks (FCN) [10] , SegNet [18] и U-Net, используют сверточные НС для изучения признаков пикселей на изображении и предсказывают класс каждого пикселя. Методы на основе регионов, такие как Region-based Convolutional Neural Network (R-CNN), Faster R-CNN [22] и Mask R-CNN, используют сверточные НС для извлечения признаков из регионов изображения и классификации каждого региона.

Семантическую сегментацию можно рассматривать, как алгоритм, выполняющий задачу автоматической классификации и выделения объектов

на изображении, а также как процесс обработки изображений, включающий применение этого алгоритма для достижения желаемых результатов.

Семантическая сегментация изображений находят практическое применение в автоматическом распознавании объектов, автономной навигации роботов, беспилотном транспорте, медицинской диагностике и многое другое. Более того, с появлением более мощных вычислительных ресурсов и развитием глубокого обучения, алгоритмы семантической сегментации стали точнее и эффективнее, что еще больше увеличило их популярность и практическую применимость.

1.2. Эволюция решений задачи сегментации

Первоначальная идея нейронной сети возникла в середине 20-го века. Первый шаг к нейронным сетям был сделан в 1943 году, когда Уоррен Маккалох и Уолтер Питтс [36] написали работу о том, как могут работать нейроны. Спустя 15 лет, нейробиолог Фрэнк Розенблатт начал работу над перцептроном [37]. Результатом его исследований стала встроенная аппаратура, которая является старейшей нейронной сетью, которая работает до сих пор. Однослойный перцептрон был признан полезным для классификации непрерывного набора входных данных на один или два класса. Перцептрон вычисляет взвешенную сумму входов, вычитает пороговое значение и выдает одно из двух возможных значений в качестве результата. В 1959 году Бернард Уидроу и Марсиан Хофф из Стэнфорда разработали модели ADALINE и MADALINE [38], которые стали первыми нейронными сетями, примененными для решения реальной задачи - адаптивной фильтрации, устраняющей эхо в телефонных линиях. Исследования нейронных сетей замерли после того, как Мински и Паперт [39] в 1969 году обнаружили пределы возможностей перцептрона.

Ключевым толчком к возобновлению интереса к обучению нейронных сетей стало повторное открытие алгоритма обратного распространения Полом

Вербосом [43], который впервые предложил использовать его для нейронных сетей. Спустя несколько лет после открытия метода Вербоса Румельхарт, Хинтон и Уильямс [44] экспериментально показали, что этот метод может генерировать внутренние представления входящих данных в скрытых слоях нейронных сетей. Алгоритм обучения был окончательно доработан Яном ЛеКуном в 1998 году, когда была опубликована работа ""Gradient-Based Learning Applied to Document Recognition" [45]. Далее про нейронные сети забыли до того момента пока GPU не стал работать быстрее чем CPU. К началу 2000-х все переключились на традиционные методы компьютерного зрения.

1.2.1. Традиционные подходы компьютерного зрения

В начале 2000-х годов для сегментации изображений широко использовались традиционные методы компьютерного зрения, такие как пороговая обработка, детекция краёв и рост областей. Эти методы сильно полагались на ручную настройку и не обладали способностью обобщать данные на разнообразные наборы данных. Производительность часто оставалась ограниченной из-за отсутствия вычислительных мощностей, особенно в сложных сценариях с запутанными фонами или неоднозначными границами объектов.

Пороговая обработка изображения — это самый простой и самый старый алгоритм для выполнения сегментации изображения. Этот метод представляет собой процесс разделения изображения на два (или более) класса пикселей, то есть на передний и задний план. Для того чтобы получить пороговое изображение, обычно исходное изображение преобразуется в а затем применяется метод порогового разделения. Этот метод также известен как бинаризация, поскольку изображение преобразуется в двоичную форму. Если значение интенсивности пикселя меньше порогового значения, то он преобразуется в 1 (белый цвет). Если значение интенсивности пикселя больше порогового значения, то пиксель преобразуется в 0 (черный).

В реализации этого очень простого алгоритма есть только одна проблема, необходимо определить оптимальный порог. В простых приложениях порог может быть установлен статически разработчиком метода. В реальном мире необходимо автоматическое определение порога. В 1979 г. Нобуюки Оцу предложил идею алгоритма, названного методом Оцу [29], который стал наиболее распространенным методом автоматического определения порога. Порог определяется путем минимизации внутриклассовой дисперсии интенсивности или, эквивалентно, максимизации межклассовой дисперсии. Применения метода Оцу представлено на рисунке 2.

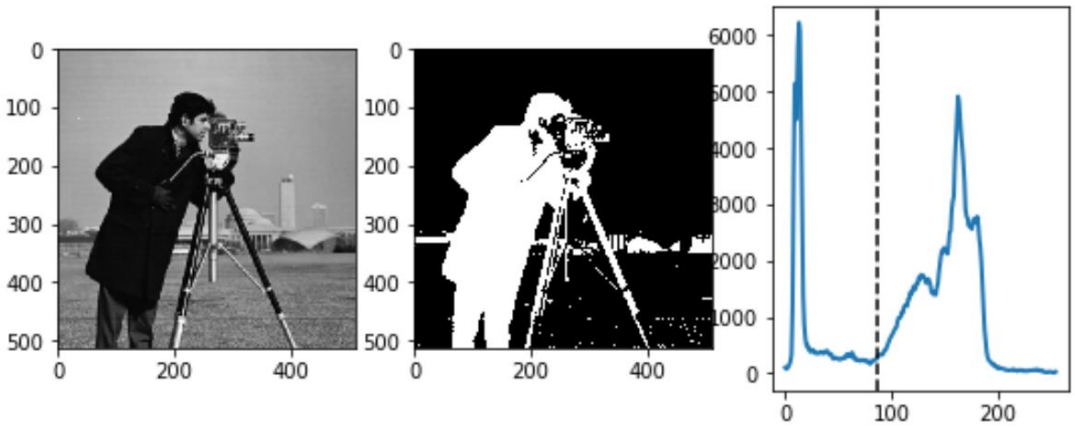

Рисунок. 2 - Пример порогового выделения по методу Оцу

Подходы, основанные на использовании случайных полей [7], являются популярным способом моделирования пространственных закономерностей в изображениях. Их применение варьируется от низкоуровневого шумоподавления до высокоуровневого распознавания объектов или категорий и полуавтоматической сегментации объектов. В ранних работах основное внимание уделялось генеративному моделированию с использованием марковских случайных полей. Модели условных случайных полей (CRF) стали более популярными благодаря их способности напрямую предсказывать сегментацию по наблюдаемому изображению. Условные случайные поля

являются эффективным инструментом для решения множества различных задач сегментации и маркировки данных, включая визуальные сегментации и маркировки данных, включая интерпретацию визуальных сцен, которая в которых ставится задача разделения изображений на составляющие их области на семантическом уровне и присвоения присвоить каждому региону соответствующие метки класса. Для точной маркировки важно улавливать глобальный контекст изображения, а также локальную информацию.

1.2.2. Свёрточные нейронные сети

Введение свёрточных НС оживило область компьютерного зрения, включая сегментацию изображений. В ранних подходах к сегментации с использованием свёрточных НС использовались архитектуры, такие как полносвязные сверточные сети. Полносвязные сверточные сети могли предсказывать пиксельные метки, заменяя полносвязные слои свёрточными, что позволяло обучать модель для задач сегментации. Однако ранние архитектуры полносвязные сверточных сетей испытывали трудности с захватом мелких деталей и страдали от проблем, таких как дисбаланс классов.

Сети "энкодер-декодер" в контексте семантической сегментации используют энкодер для сжатия входного изображения в латентное представление, содержащее основную семантическую информацию, и декодер для восстановления изображения из этого представления с целью предсказать карту сегментации. Они обычно включают соединения между слоями энкодера и декодера, чтобы передать пространственную информацию при восстановлении изображения.

Например, в DeConvNet энкодер состоит из последовательности операций пуллинга и свёртки, в то время как декодер, построенный сверху, увеличивает размер изображения с помощью операций распуллинга и деконволюции [30]. Max-пуллинговые позиции сохраняются на каждом

уровне энкодера и передаются в соответствующие уровни декодера. Все это переросло в архитектуру U-Net.

Архитектура U-Net, представленная в 2015 году, решала ограничения ранних подходов полносвязные сверточных сетей к сегментации. U-Net ввела пропускные соединения между кодировщиком и декодировщиком, облегчая восстановление пространственной информации, потерянной при уменьшении размера [3]. Эта архитектура значительно улучшила точность сегментации, особенно в медицинских задачах обработки изображений, таких как сегментация клеток и обнаружение опухолей и является популярнейшей архитектурой на сегодняшний момент.

UNet++, предложенная как усовершенствование оригинальной архитектуры UNet, включает дополнительные промежуточные уровни, называемые nested skip pathways. Эти уровни обеспечивают более плотное и плавное соединение между аналогичными уровнями кодировщика и декодировщика, что позволяет более эффективно передавать и сохранять пространственную информацию на протяжении всей сети [35]. Такая структура улучшает способность модели к распознаванию сложных и мелких объектов, что особенно важно для задач, требующих высокой точности сегментации, таких как медицинская диагностика и обработка спутниковых изображений.

Использование dense convolutional blocks (DCB) в промежуточных уровнях UNet++ способствует улучшению передачи информации и снижению эффекта градиентного затухания, что позволяет модели более эффективно обучаться на сложных данных. Несмотря на повышенные требования к вычислительным ресурсам и памяти по сравнению с оригинальным UNet, UNet++ демонстрирует значительные улучшения в точности и качестве сегментации.

DeepLab — это архитектура, разработанная для сегментации изображений на основе глубокого обучения. Она была разработана для улучшения точности сегментации объектов путем использования различных

передовых технологий. Основные компоненты и особенности архитектуры DeepLab:

- DeepLab использует сверточные нейронные сети (CNN) в качестве базовой сети для извлечения признаков. Среди популярных архитектур базовой сети — ResNet, Xception и MobileNet;
- дилатированная свертка используется для увеличения поля зрения сверточных слоев без увеличения количества параметров и вычислительной сложности. Это достигается путем введения "дыр" (нулевых значений) между элементами ядра свертки. Это позволяет сети захватывать информацию на большем масштабе, сохраняя при этом детализированную информацию;
- ASPP (Atrous Spatial Pyramid Pooling) [5] — это ключевой компонент DeepLab, который использует несколько атрозных сверток с разными коэффициентами расширения для захвата контекстной информации на разных масштабах. ASPP включает параллельные пути с различными уровнями атрозной свертки [6], а также глобальную усреднительную пулизацию для объединения глобального контекста;
- Эффективный декодер - в более поздних версиях DeepLab, таких как DeepLabv3+, добавляется декодер, который улучшает детализацию сегментации на границах объектов [13]. Декодер объединяет высокоуровневые признаки из ASPP с более детализированными низкоуровневыми признаками из базовой сети через механизм skip connections [4].

Mask R-CNN [19] — это расширение Faster R-CNN, предназначенное для решения задач объектной детекции и сегментации на уровне пикселей (instance segmentation). Эта архитектура объединяет задачи детекции объектов и сегментации, позволяя не только находить объекты в изображении, но и определять их точные границы. Основной компонент Mask R-CNN — базовая сверточная нейронная сеть, такая как ResNet или ResNeXt, которая

используется для извлечения признаков из входного изображения. Также применяется сеть Feature Pyramid Network [20] для улучшения качества признаков на разных масштабах.

Ключевой элемент Mask R-CNN — Сеть региональных предложений (Region Proposal Network, RPN), которая генерирует потенциальные регионы интереса (RoI) в изображении, где могут находиться объекты. Эти регионы представляют собой прямоугольные области, оцениваемые по вероятности наличия объекта. Для более точного выравнивания признаков используется RoI Align, который устраняет квантование, происходящее при RoI Pooling, сохраняя пространственную точность.

Mask R-CNN имеет несколько подсетей для выполнения разных задач. Классификационная подсеть определяет, к какому классу относится объект в каждом RoI, тогда как подсеть регрессии уточняет координаты ограничивающего прямоугольника для каждого RoI. Важной частью является подсеть маски, которая генерирует побитовую маску для каждого объекта, что позволяет выделить точные границы объекта на уровне пикселей.

Процесс работы Mask R-CNN начинается с извлечения признаков из входного изображения с помощью базовой сети. Затем RPN генерирует множество потенциальных регионов интереса, которые передаются в RoI Align для точного извлечения признаков. Подсети классификации и регрессии определяют классы объектов и уточняют их координаты, в то время как подсеть маски генерирует точные маски для каждого объекта.

В итоге, Mask R-CNN — это мощная и универсальная архитектура для детекции и сегментации объектов на уровне пикселей. Она объединяет возможности Faster R-CNN для детекции объектов с компонентом для сегментации масок, что позволяет точно определять границы объектов. Использование RoI Align и других технологий обеспечивает высокую точность и детализацию в процессе обработки изображений.

EfficientNet — это семейство сверточных нейронных сетей, разработанных для достижения высокого уровня точности при минимизации

вычислительных затрат [31]. Основная идея заключается в сбалансированном увеличении трех ключевых параметров сети: глубины, ширины и разрешения. EfficientNet использует методику, называемую "compound scaling", которая позволяет масштабировать эти параметры совместно и эффективно.

Архитектура EfficientNet базируется на первоначально предложенной модели EfficientNet-B0, которая затем масштабируется до более крупных моделей, таких как EfficientNet-B1, B2 и так далее, до B7. Каждая из этих моделей увеличивает глубину, ширину и разрешение входного изображения сбалансированным образом, что позволяет улучшить точность модели при относительно низких вычислительных затратах. EfficientNet достигла значительных успехов в различных задачах компьютерного зрения благодаря своему эффективному подходу к масштабированию.

В основе EfficientNet лежит архитектура MobileNetV2 с дополнительными усовершенствованиями, такими как свертки с использованием смешанных сверточных групп и свертки с расширением. Эти улучшения способствуют более эффективному извлечению признаков и уменьшению количества параметров модели, что делает EfficientNet одной из самых производительных архитектур CNN на момент её создания.

EfficientSeg — это архитектура для сегментации изображений, разработанная на основе принципов EfficientNet для достижения высокой производительности при низких вычислительных затратах [32]. EfficientSeg использует преимущества EfficientNet для извлечения признаков и включает дополнительные компоненты, специфичные для задачи сегментации.

Основной компонент EfficientSeg — это декодер, который преобразует высокоуровневые признаки, извлеченные из базовой сети, обратно в изображение с той же размерностью, что и входное изображение, но с соответствующими метками для каждого пикселя. Декодер EfficientSeg обычно включает механизмы, такие как upsampling (увеличение масштаба) и skip connections (пропуски соединений), которые помогают восстановить пространственную информацию и улучшить точность сегментации.

Процесс работы EfficientSeg начинается с того, что входное изображение пропускается через базовую сеть EfficientNet, которая извлекает признаки. Эти признаки затем передаются в декодер, который выполняет upsampling и восстанавливает пространственные характеристики для создания сегментированного изображения. Skip connections позволяют передавать информацию из ранних слоев базовой сети в соответствующие слои декодера, что помогает сохранять детали и улучшает сегментацию.

В результате, EfficientSeg сочетает эффективность и точность EfficientNet с возможностями сегментации, создавая мощную и производительную архитектуру для задач сегментации изображений. EfficientSeg демонстрирует высокую точность при низких вычислительных затратах, что делает её подходящей для применения в различных областях.

1.2.3. Трансформеры для сегментации изображений

Адаптация архитектур трансформеров, в основном разработанных для задач обработки естественного языка, к сегментации изображений, отметила значительный сдвиг в этой области. Модели, такие как ViT, Swin [25] и SegFormer [24] продемонстрировали конкурентоспособную производительность на бенчмарках по сегментации. Модели сегментации на основе трансформеров используют механизмы СВ для захвата долгосрочных зависимостей и глобального контекста, что привело к улучшению точности сегментации.

Сравнение этих архитектур на бенчмарках, таких как COCO [34], PASCAL VOC [33] и Cityscapes [8], показывает явную тенденцию к улучшению производительности со временем. В то время как ранние методы боролись в основном с точностью и обобщением, новые достижения в области архитектур глубокого обучения расширили границы задач сегментации, достигая замечательных результатов даже в сложных сценариях. Кроме того, эволюция в сторону более эффективных моделей подчеркивает важность

масштабируемости и оптимизации ресурсов в прикладных задачах реального мира.

1.3. Выбор инструмента

TensorFlow, PyTorch— это 2 наиболее популярных инструмента для глубокого обучения, которые могут быть использованы для решения задачи семантической сегментации.

TensorFlow — это открытая программная библиотека для машинного обучения, разработанная компанией Google. Она предназначена для построения и обучения различных моделей искусственного интеллекта, включая НС. TensorFlow предоставляет гибкие инструменты для создания и обучения разнообразных моделей машинного обучения, а также удобные средства для развертывания и использования этих моделей на различных платформах, включая сервера, мобильные устройства и веб-приложения. Он широко используется как в академической сфере, так и в промышленности для решения задач анализа данных, распознавания образов, обработки естественного языка, компьютерного зрения и других. TensorFlow поддерживает несколько языков программирования, в том числе Python, C++, Java и Go.

PyTorch — это библиотека для машинного обучения и глубокого обучения [40]. Она разрабатывается командой Facebook's AI Research (FAIR). PyTorch предоставляет гибкий и удобный интерфейс для построения и обучения различных моделей глубокого обучения, включая НС. Одним из ключевых преимуществ PyTorch является его динамический вычислительный граф, который делает процесс создания и обучения моделей более гибким и интуитивно понятным. PyTorch также предлагает богатый набор инструментов для исследования, прототипирования и развертывания моделей машинного обучения, а также поддерживает распределенное обучение и работу на графических процессорах (GPU) для ускорения процесса обучения.

В качестве инструмента был выбран PyTorch, т.к. он имеет большую гибкость, а также множество реализованных архитектур на github и paperwithcode. PyTorch предоставляет простой и интуитивно понятный интерфейс, имеет мощную поддержку GPU.

Segmentation Models PyTorch – библиотека, основанная на фреймворке PyTorch и предназначенная для упрощения процесса создания, обучения и использования моделей глубокого обучения для сегментации объектов на изображениях [42].

Все модели запускались на платформе Kaggle. Kaggle - одна из крупнейших и самых популярных платформ для соревнований по анализу данных и машинному обучению. Выбор Kaggle в качестве основного инструмента обусловлен несколькими ключевыми преимуществами:

- Инструменты и ресурсы: Kaggle предоставляет бесплатные вычислительные ресурсы, включая использование GPU, что важно для обучения моделей глубокого обучения;
- Создание собственных датасетов: Kaggle позволяет загружать огромные объемы данных, доступ к которым можно получить откуда угодно;
- Версионирование: новый функционал в блокнот можно добавлять под новую версию и всегда иметь возможность вернуться к старой.

1.4. Архитектуры нейронных сетей

1.4.1. U-Net

U-Net — это тип архитектуры сверточной НС, который был разработан в 2015 году для задач биомедицинской сегментации изображений. Особенностью U-Net является ее уникальная "U"-образная структура, которая позволяет точно сегментировать изображения. Состоит из двух основных частей — сжимающего пути (энкодера) и расширяющего пути (декодера). Архитектура U-Net представлена на рисунке 3.

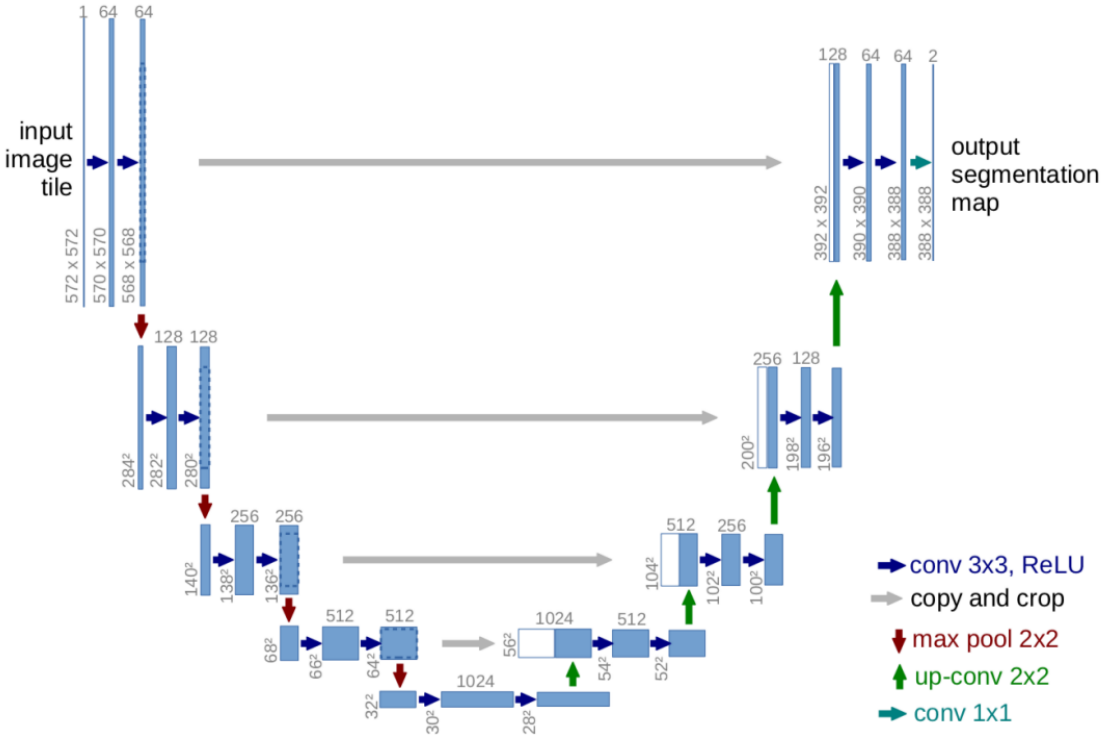

Рисунок 3 - Архитектура U-Net

Энкодер последовательно уменьшает размеры пространственных данных изображения, увеличивая при этом количество признаковых каналов. Он состоит из нескольких блоков, каждый из которых включает в себя сверточные слои, функцию активации (обычно ReLU) и слой пулинга (чаще всего max pooling), что помогает извлекать более абстрактные признаки изображения.

Декодер постепенно увеличивает пространственные размеры данных, одновременно уменьшая количество признаковых каналов. В каждом блоке декодера происходит апсемплинг (или транспонированная свертка) предыдущего слоя, за которым следует объединение с соответствующими признаками из энкодера (операция "skip-connection") и свертка. Это позволяет восстанавливать детали изображения и точно локализовать области интереса.

Важной особенностью U-Net являются соединения пропуска, которые напрямую соединяют блоки энкодера с соответствующими блоками декодера.

Эти соединения помогают передавать контекстную информацию из энкодера в декодер, что улучшает точность локализации при сегментации.

1.4.2. Attention U-Net

Архитектура Attention U-Net [21] представляет собой модификацию классической U-Net с добавлением МВ. Она состоит из двух основных компонентов: энкодера и декодера. Энкодер представляет собой сверточную НС, которая постепенно уменьшает размерность входного изображения и извлекает его признаки на разных уровнях абстракции. Декодер, в свою очередь, использует блоки внимания для восстановления высокоуровневых деталей и контекста в сегментированном изображении. Архитектура сети Attention U-Net представлена на рисунке 4.

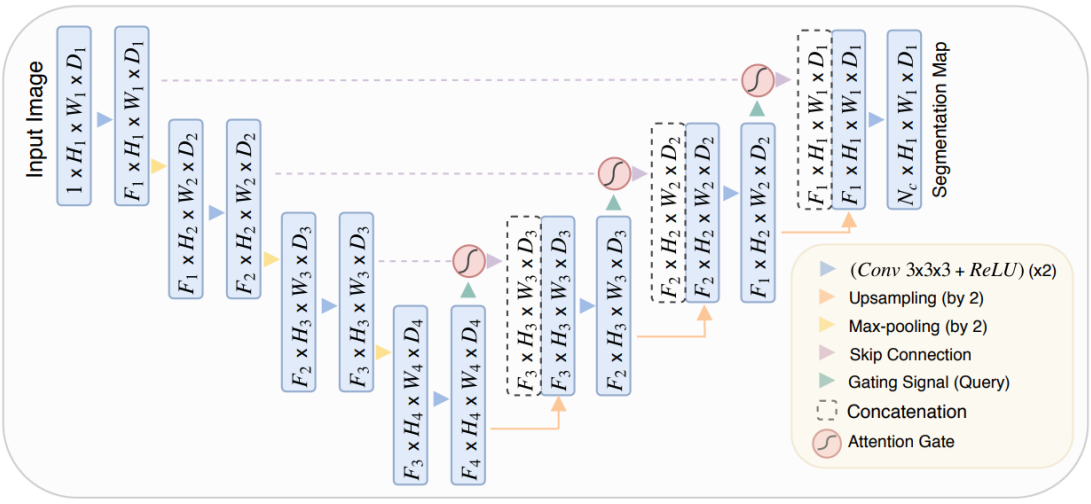

Рисунок 4 - Архитектура Attention U-Net

МВ в Attention U-Net является ключевым элементом для увеличения точности семантической сегментации изображений. Он позволяет сети акцентировать внимание на важных частях изображения и учитывать контекстуальную информацию для более точного распознавания объектов. Процесс создания МВ описан в статье Стефании Кристины [11].

По сравнению с оригинальной моделью U-Net, Attention U-Net значительно улучшает производительность. Фокусировка на наиболее значимых областях изображения позволяет добиться более точной сегментации и уменьшить воздействие шума и несущественных объектов.

Однако внедрение MB также повышает сложность модели, что может сделать ее более затратной с точки зрения вычислений и сложнее в обучении. Поэтому важно сбалансировать преимущества увеличения производительности с увеличенными вычислительными требованиями.

1.4.3. SegFormer

SegFormer — это современная модель для семантической сегментации изображений, которая объединяет методы внимания и трансформера для эффективной обработки изображений.

Основной концепцией SegFormer является применение MB трансформера для работы с пиксельными данными в задаче сегментации. Это позволяет модели учитывать контекстуальные связи между пикселями и улучшить точность сегментации. Архитектура модели представлена на рисунке 5.

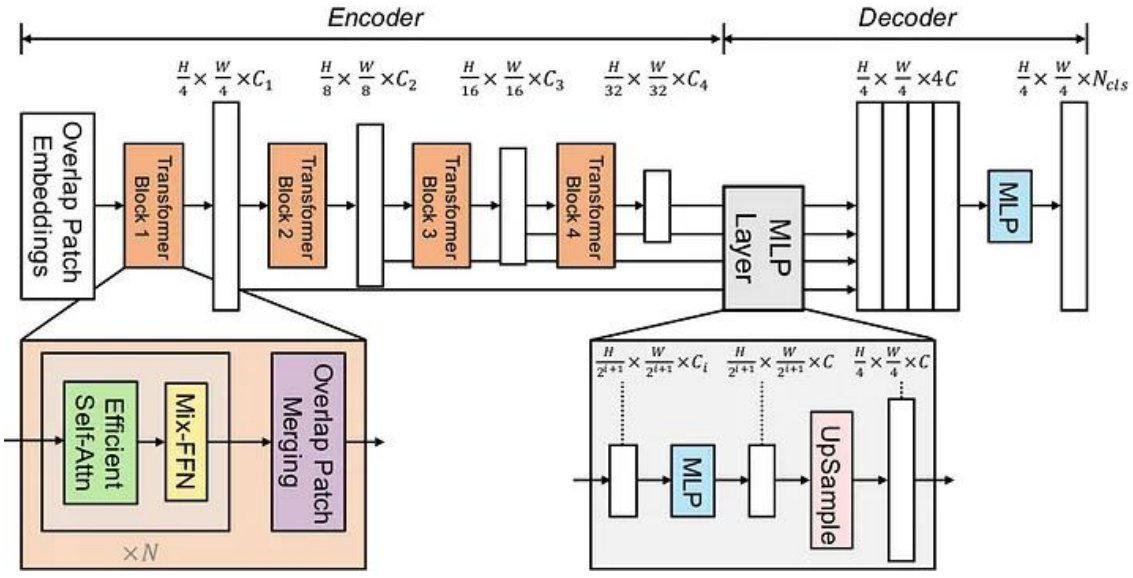

Рисунок 5 – Архитектура SegFormer

Основные компоненты модели SegFormer:

1) Overlap Patch Embeddings (вложение перекрывающихся частей) - слой, используемый в модели SegFormer, который выполняет преобразование перекрывающихся частей изображения в компактные векторные представления, называемые эмбеддингами. Работа этого слоя включает следующие шаги:

- Разделение изображения на патчи: Исходное изображение разбивается на набор перекрывающихся патчей фиксированного размера. Это позволяет модели обрабатывать изображения различных размеров и масштабов.
- Embedding из патчей: Каждый патч изображения проходит через предварительно обученный векторный Embedding слой. Этот слой преобразует каждый патч в векторное представление, которое содержит информацию о содержании и структуре патча.
- Слияние Embeddings патчей: Полученные Embeddings из различных патчей объединяются и суммируются с помощью механизма слияния. Это может включать в себя операции суммирования, усреднения или объединения Embeddings для создания обобщенного представления всего изображения.
- Получение итогового представления: в результате работы слоя Overlap Patch Embeddings получается итоговое векторное представление всего изображения, которое содержит информацию о структуре и содержании всех его частей.

Этот процесс помогает модели SegFormer эффективно обрабатывать изображения и извлекать компактное и информативное представление, которое затем может быть использовано для сегментации объектов на изображении.

2) Efficient Self-Attention - слой в модели SegFormer, который играет ключевую роль в обработке изображений и извлечении контекстуальных зависимостей между пикселями. Работа Efficient Self-

Attention слоя начинается с процесса вычисления трех матриц: запроса (query), ключа (key) и значения (value). Запрос и ключ используются для вычисления весового коэффициента внимания между различными позициями, а значения используются для формирования агрегированных представлений.
Далее, с помощью произведения запроса и ключа, и применения функции Softmax получаются весовые коэффициенты внимания. Эти коэффициенты определяют важность каждой позиции в данных при вычислении агрегированных представлений. Затем, с учетом весового коэффициента, значения умножаются на соответствующие веса и суммируются для получения агрегированных представлений. Результатом работы Efficient Self-Attention слоя являются агрегированные представления, которые содержат информацию о взаимосвязях и зависимостях между позициями во входных данных.

3) Mix-FNN (Mix-Feature Neuron Network) — это слой, используемый в модели SegFormer, который выполняет смешивание признаков разных уровней для получения более выразительных и информативных представлений.

Работа слоя Mix-FNN в SegFormer включает следующие шаги:

- Получение признаков из разных уровней абстракции: После прохождения изображения через блоки трансформера, в модели SegFormer получаются признаки разного уровня абстракции, каждый из которых содержит информацию о различных аспектах изображения.
- Объединение признаков из разных источников: Mix-FNN принимает на вход признаки из разных уровней абстракции, полученные на различных этапах обработки изображения. Это позволяет модели комбинировать информацию из различных источников, что может быть полезным для лучшего понимания структуры объектов на изображении.

— Агрегация признаков: после объединения признаков из разных источников, Mix-FNN агрегирует эти признаки для создания более полного и обобщенного представления изображения. Это может включать в себя операции суммирования, конкатенации или применения различных функций активации для улучшения представления признаков.

— Обработка смешанных признаков: на последнем этапе, Mix-FNN обрабатывает смешанные признаки с использованием полносвязных НС. Это позволяет модели выполнить дополнительные преобразования и выделить более важные признаки для улучшения качества сегментации.

Таким образом, слой Mix-FNN в SegFormer играет важную роль в комбинировании и агрегации информации из разных уровней абстракции для создания более эффективного представления изображения, что способствует более точной сегментации объектов.

4) Overlap Patch Merging (слияние перекрывающихся патчей) — это важный компонент модели SegFormer, который объединяет перекрывающиеся части изображения для создания иерархической карты признаков на разных уровнях. Этот процесс позволяет получать многоуровневые признаки различных масштабов, что способствует более эффективному извлечению и использованию информации о различных объектах на изображении.

Работа слоя Overlap Patch Merging включает несколько этапов.

— Разделение изображения на патчи: Входное изображение разделяется на набор перекрывающихся патчей фиксированного размера.

— Извлечение признаков из патчей: Каждый патч изображения проходит через предварительно обученную модель сверточной НС для извлечения признаков. Это позволяет получить набор признаков

для каждого патча, который содержит информацию о структуре и содержании объектов в них.

— Слияние и агрегация признаков: Полученные признаки из различных патчей объединяются и агрегируются с помощью механизма слияния. Это может включать в себя операции суммирования, объединения или усреднения признаков для создания более обобщенного представления.

— Формирование иерархической карты признаков: в результате работы слоя Overlap Patch Merging формируется иерархическая карта признаков, где каждый уровень содержит признаки различного масштаба и содержит информацию о различных деталях и структурах объектов на изображении.

Многослойный перцептрон (MLP) в модели SegFormer представляет собой сетевой слой, который осуществляет нелинейное преобразование входных данных. Он состоит из нескольких последовательных полносвязных слоев, где каждый слой содержит набор нейронов с нелинейной функцией активации. В процессе работы MLP слой получает на вход данные, представленные в виде вектора или матрицы, и проходит через несколько полносвязных слоев. Каждый слой применяет линейное преобразование к входу, а затем применяет нелинейную функцию активации, такую как ReLU (Rectified Linear Unit), для внесения нелинейности в данные. Таким образом, MLP слой генерирует выходные данные, которые представляют собой преобразованные и более абстрактные представления исходных признаков.

В заключение, модель SegFormer имеет ряд преимуществ, таких как эффективное использование контекстуальных зависимостей с помощью механизма трансформера, способность работать с объектами различного масштаба и возможность использования самообучения и предварительного обучения для улучшения качества сегментации.

2. Исследование набора данных и анализ метрик

2.1. Подготовка набора данных

Набор данных собран из 1660 снимков [2], сделанных в карьере в ходе 11 экспериментов в весеннее, летнее, осеннее и зимнее время и в разную погоду. Каждый эксперимент содержал изображения кусков породы из 3 - 5 мест отработки карьера, выбранных геологической службой компании, занимающейся переработкой месторождений, как наиболее репрезентативные. Файлы меток имеют формат COCO. Все изображения имеют разрешение 2590 x 2048 пикселей. Разрешение около 4 пикселей в 1 мм на расстоянии 5 м, что считается достаточным по сравнению с типичной шириной асбестовых прожилок (около 4-12 мм). Пример изображения с разметкой представлен на рисунке 6.

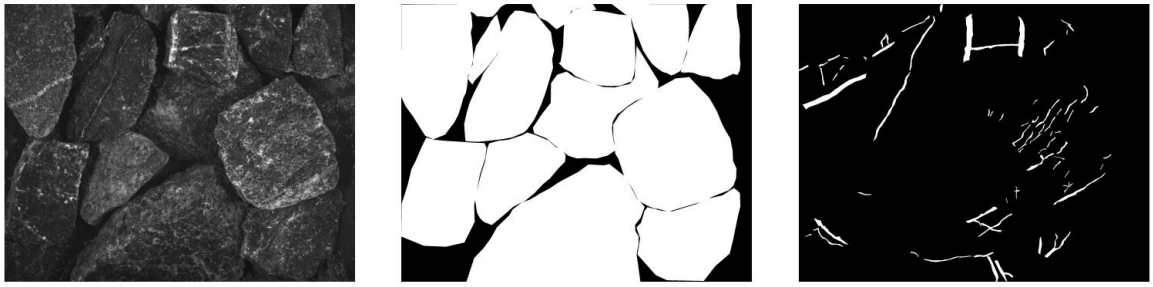

Рисунок 6 - изображение, разметка для камней и прожилок

При первом рассмотрении выяснилось, что в наборе данных имеются полная разметка присутствует не для каждого изображения. Она должна в себя включать 2 класса: камень и асбестовые прожилки. Первым этапом был выбор только изображений с полной разметкой. Из 1660 изображений было отобрано 1169.

Следующей проблемой было наличие дубликатов и изображений с плохо выделенной разметкой. Решением было либо полное удаление изображения и дубликатов, либо удаление дубликатов и оставление маски,

наиболее четко описывающую изображение. Изображения с неправильной или неполной разметкой просто удалялись из датасета. В ходе второго этапа фильтрации из 1169 изображения было отобрано 534.

Для возможности обработки моделью изображения были уменьшены до размера 512x512. Данные были разделены на тренировочную и тестовую выборку в соотношении 80% на 20%. Тренировочные данные были подвержены аугментации с помощью библиотеки Albumentations [15], а конкретно были увеличены, рандомно обрезаны, повернуты и искажены. Над тестовой выборкой никаких манипуляций произведено не было для возможности сравнения результатов разных моделей и функций потерь.

2.2. Анализ метрик для задачи семантической сегментации

Прежде чем обучать модель необходимо определить метрики, которые будут использоваться для обучения и для проверки правильности обучения. Стоит иметь ввиду, что прожилки имеют маленькую площадь. Это значит, что мы имеем ситуацию, в которой есть явный перекос в сторону класса фона. Подробно метрики разобраны в статье E. Tiu [14] и Jeremy Jordan [16].

1) Матрица ошибок (Confusion Matrix). Матрица ошибок представляет собой таблицу, показывающую распределение истинных и предсказанных классов. В случае бинарной сегментации она включает:

- True Positives (TP): Количество пикселей, правильно отнесенных к положительному классу.
- True Negatives (TN): Количество пикселей, правильно отнесенных к отрицательному классу.
- False Positives (FP): Количество пикселей, ошибочно отнесенных к положительному классу.
- False Negatives (FN): Количество пикселей, ошибочно отнесенных к отрицательному классу.

2) Точность (Accuracy). Точность показывает долю правильно классифицированных пикселей, но при сильном дисбалансе классов может быть введущей в заблуждение, так как большинство правильных классификаций может быть обусловлено преобладающим классом.

3) Полнота (Recall). Полнота показывает, какую долю пикселей положительного класса модель правильно классифицировала. Важно учитывать для анализа способности модели находить пиксели редкого класса.

4) Точность (Precision). Точность показывает, какую долю предсказанных моделью положительных пикселей действительно являются положительными.

5) Индекс Джаккарда (Jaccard Index). Индекс Джаккарда, или коэффициент пересечения, измеряет сходство между предсказанным и истинным набором пикселей положительного класса [12]. Хорошо подходит для сегментационных задач с дисбалансом классов.

6) Коэффициент Дайса (Dice Coefficient). Коэффициент Дайса близок к индексу Джаккарда, но придает больше веса пересекающимся элементам.

7) Бинарная кросс-энтропия (Binary Cross-Entropy, BCE). BCE используется в задачах, где нужно предсказать вероятность принадлежности каждого примера к одному из двух классов. Из преимуществ можно выделить хорошую сходимость, основанную на принципах теории информации и максимального правдоподобия, легкость вычисления и широкую нативную поддержку библиотеками машинного обучения. Из недостатков можно отметить плохую эффективность при сильном дисбалансе классов, так как модель может быть склонна предсказывать преобладающий класс, игнорируя редкий класс, и малую интерпретируемость штрафов. Для решения проблем дисбаланса классов применяют усложненные функции потерь такие как взвешенная бинарная кросс энтропия или Фокальную функцию потерь.

8) Фокальный лосс (Focal Loss). Специально разработанная для задач с сильным дисбалансом классов функция потерь. Фокальный лосс модифицирует стандартную кросс-энтропию, добавляя взвешивающий коэффициент, который уменьшает вклад легко классифицируемых примеров и увеличивает вклад трудно классифицируемых примеров. Это позволяет модели уделять больше внимания сложным для классификации пикселям или объектам.

9) Лосс Тверского (Tversky Loss). Tversky Loss основан на коэффициенте Тверского, который является обобщением индексов Дайса и Джаккарда [41]. Он вводит параметры для регулировки вклада ложно положительных и ложно отрицательных результатов, что особенно полезно для задач сегментации с дисбалансом классов.

10) Тверский фокальный лосс (TverskyFocalLoss). TverskyFocalLoss объединяет идеи фокального лосса и Tversky Loss для создания функции потерь, которая одновременно учитывает дисбаланс классов и фокусируется на трудно классифицируемых примерах. Он предоставляет более гибкий и адаптивный подход, за счет использования параметров a и b, которые позволяют компенсировать дисбаланс классов, регулируя вклад FP и FN, и фокусировки на трудных примерах используя фокусирующий параметр γ.

2.3. Выбор метрик для задачи определения асбестовых прожилок

После определения задачи, используемых моделей и выбора базы данных для обучения работы моделей, выберем метрики для оценки качества работы и обучения НС.

В результате анализа метрик было выявлено следующее:

— BCELoss учитывает ложные срабатывания (FP), поэтому сам по себе это не очень хороший показатель для задач сегментации с большим количеством фона (как здесь).

─ Dice (не учитывает FP) — хороший показатель для несбалансированных наборов данных.

─ Комбинированный лосс DiceBCELoss (Dice + взвешенный BCE) с небольшим весом bce — хороший способ быстро уменьшать ошибку — используя хорошую сходимость от BCE и концентрируясь на Dice.

─ Еще один хороший способ — использовать TverskyFocalLoss, который сосредотачивает внимание на небольших масках сегментации и помогает бороться с проблемой дисбаланса классов.

В качестве используемых функций потерь выбраны:

─ DiceBCELoss

─ TverskyFocalLoss

3. Тестирование нейросетевых алгоритмов

Для проведения экспериментов были реализованы на языке Python с использованием библиотеки PyTorch оригинальные нейросетевые модели: U-Net, U-Net(efficientnet-b5 encoder), Attention U-Net ,SegFormer. Модели были выбраны на основании анализа из популярности и производительности [27].

Модели обучались на платформе Kaggle с использованием GPU T4. Обучение моделей производилось на протяжении 30 эпох. Такое мало количество выбрано из-за размера датасета и использования уже предобученных моделей.

3.1. UNET

Сначала была протестирована стандартная сеть U-Net. В качестве функции потерь была выбрана DiceBCE, которая варьировала значение коэффициента с 1 (превращается в функцию Дайса) до 0,01. Результаты сегментации с помощью UNET представлены на рисунке 7.

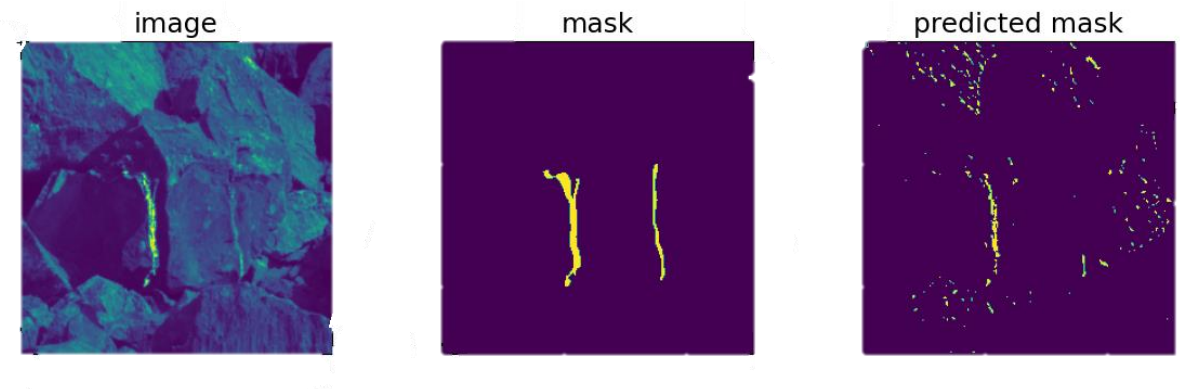

Рисунок 7. Результат работы стандартной модели Unet.

Из результатов на изображениях заметно, что модель предсказывает, но делает это не так точно, как хотелось бы. Метрика BCE равна 0.109, а коэффициент Дайса меньше 5% на валидационных данных.

3.2. UNET с энкодером efficientnet-b5

После неудачного использования стандартной сети U-Net. Было принято использовать предобученную сеть на датасете imagenet [26] с энкодером efficientnet-b5. Данная сеть тестировалась с разными функциями потерь. Результаты приведены в таблице 1 и на рисунках 8-19.

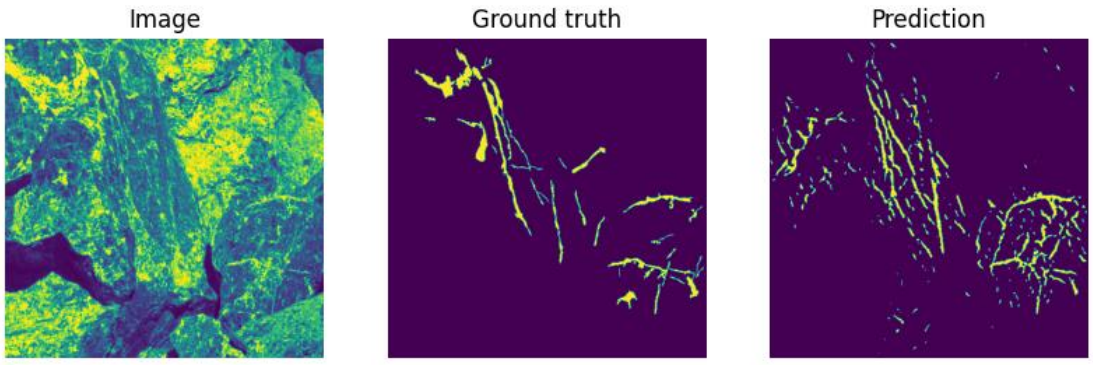

Рисунок 8 - Предобученная U-Net с DiceBCELoss, коэфициент BCE=1

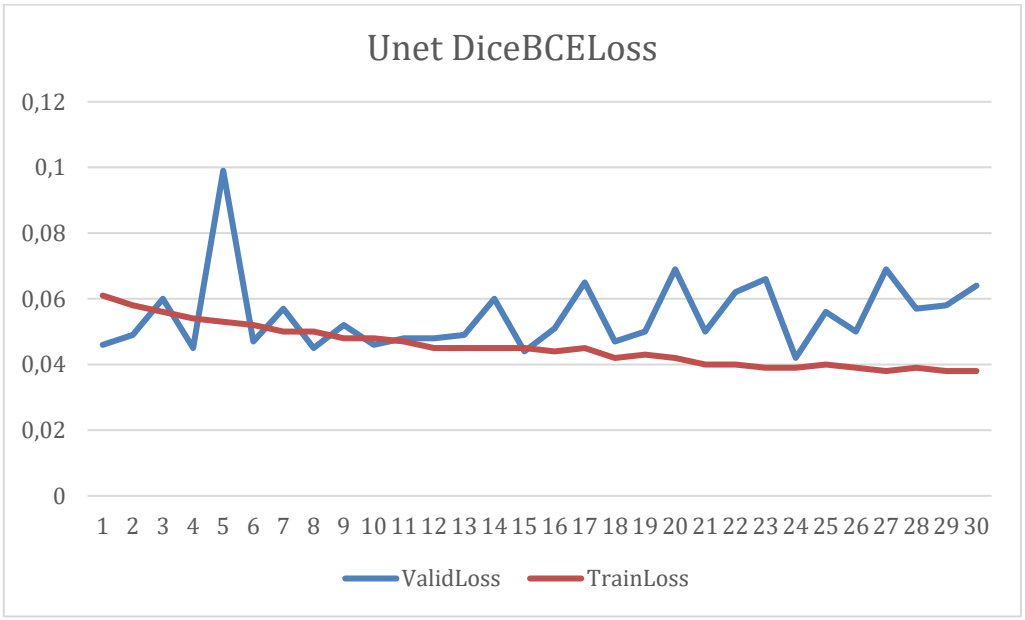

Рисунок 9 - Тренировочные кривые для U-Net, BCE=1, DiceBCELoss

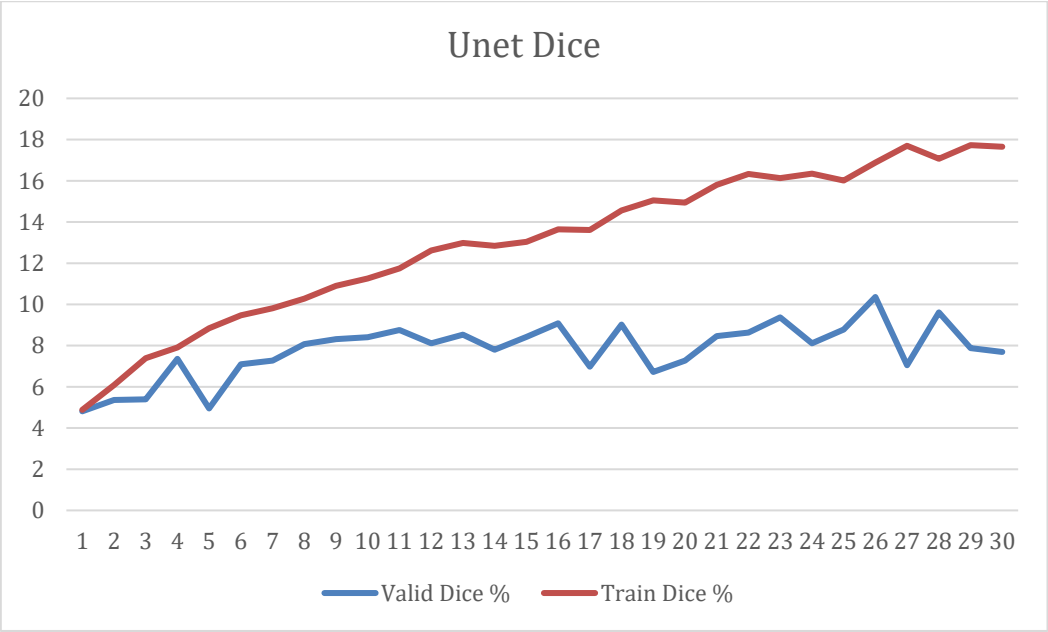

Рисунок 10 - Тренировочные кривые для U-Net, BCE=1, коэфициент Дайса

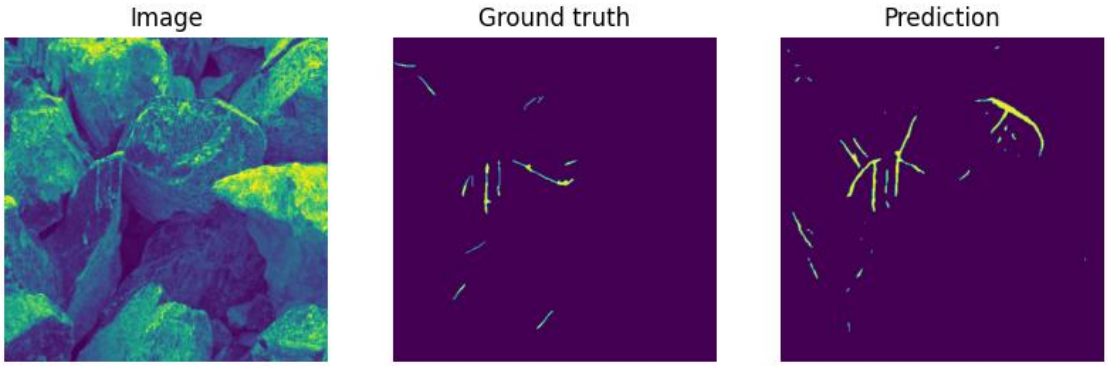

Рисунок 11 - предобученная U-Net с DiceBCELoss, коэфициент BCE=0.1

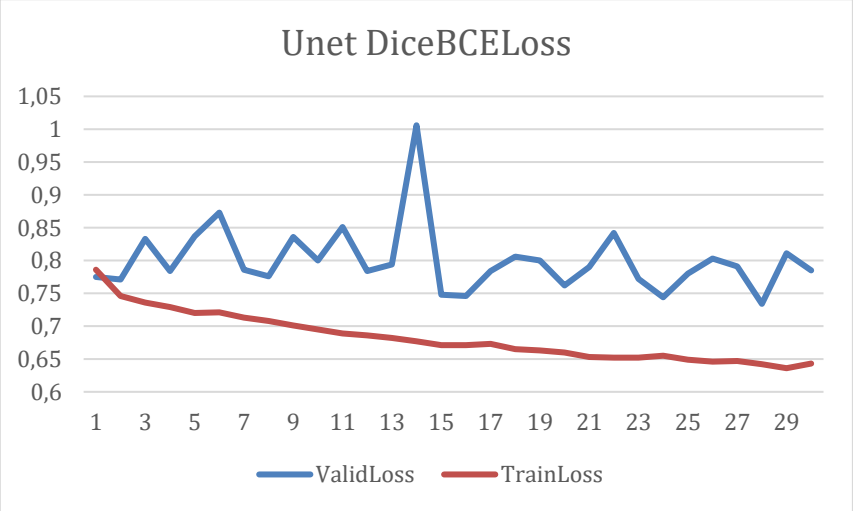

Рисунок 12 - Тренировочные кривые для U-Net, BCE=0.1, DiceBCELoss

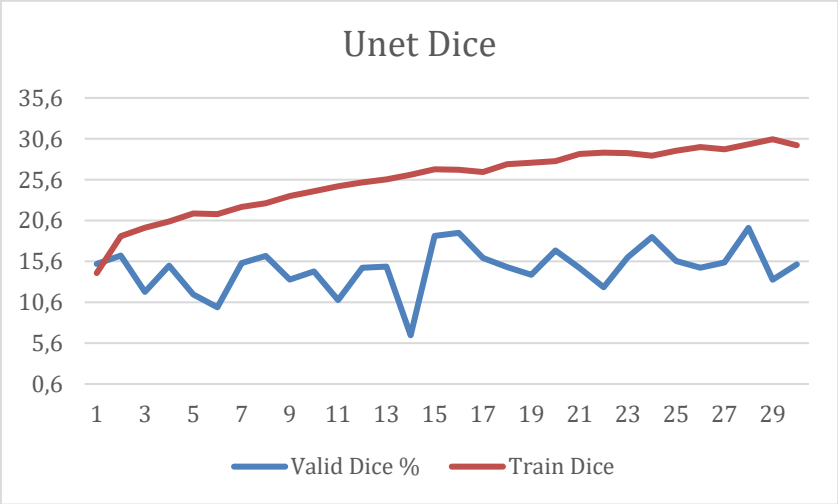

Рисунок 13 - Тренировочные кривые для U-Net, BCE=0.1, коэфициент Дайса

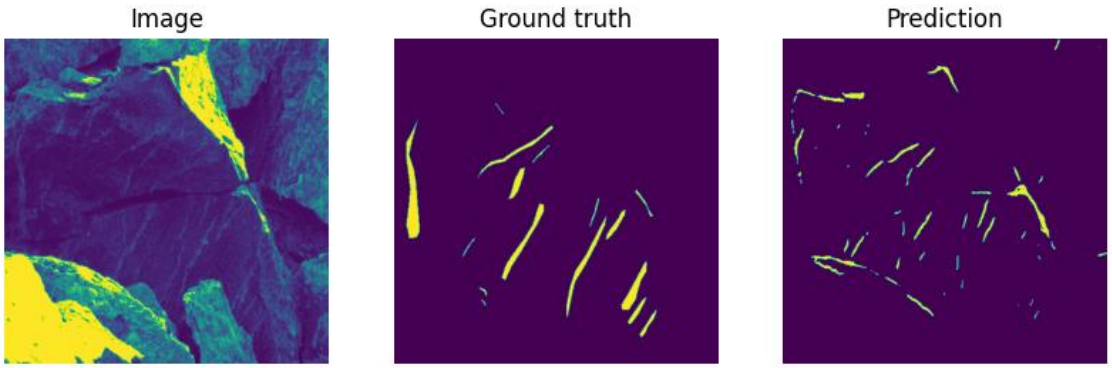

Рисунок 14 - предобученная U-Net с DiceBCELoss, коэфициент BCE=0.01

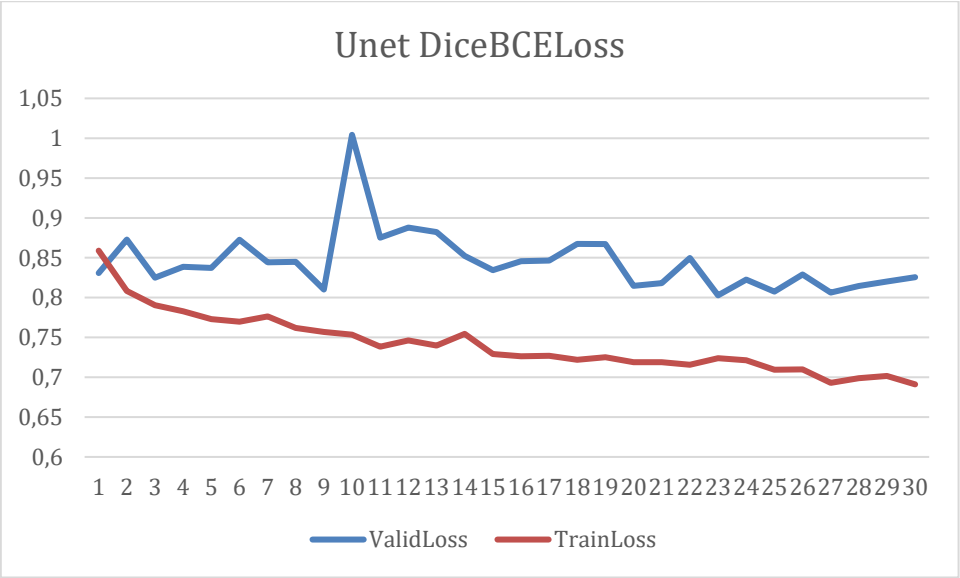

Рисунок 15 - Тренировочные кривые для U-Net, BCE=0.01, DiceBCELoss

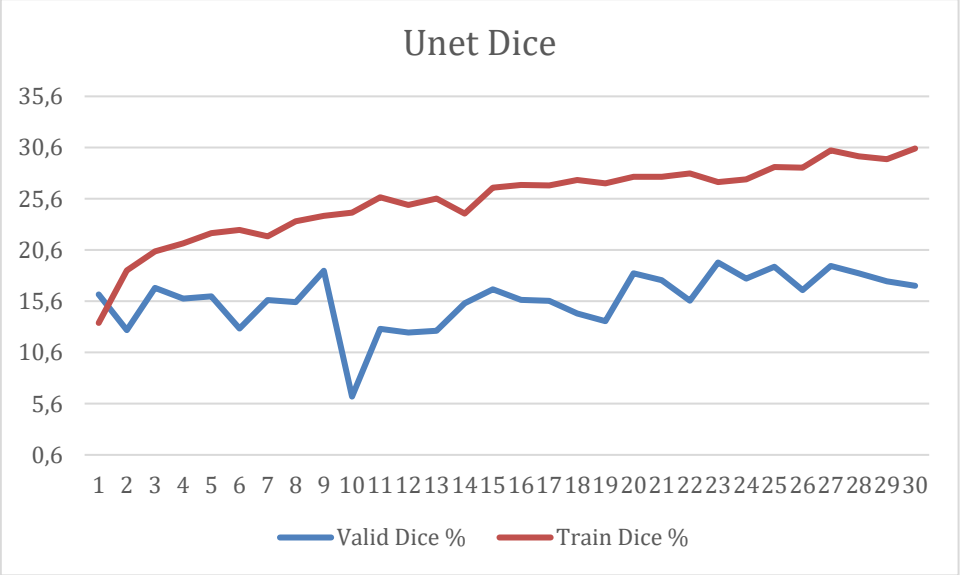

Рисунок 16 - Тренировочные кривые для U-Net, BCE=0.01, коэфициент Дайса

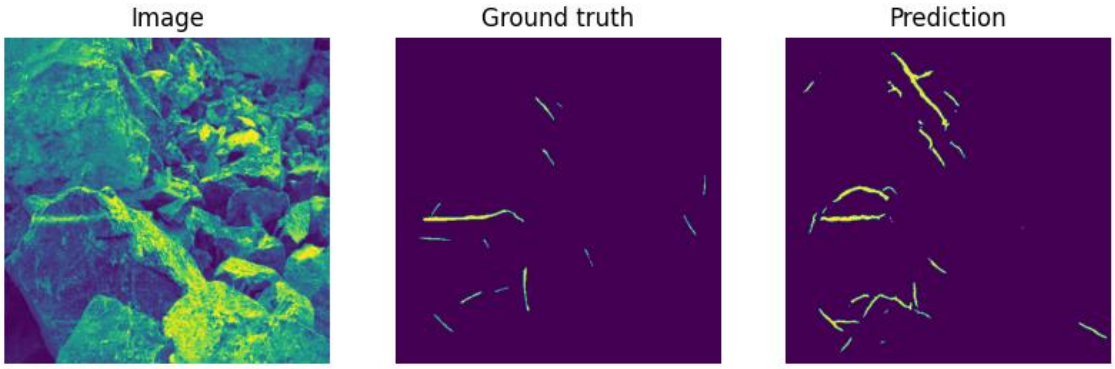

Рисунок 17 - Предобученная U-Net с TverskyFocalLoss

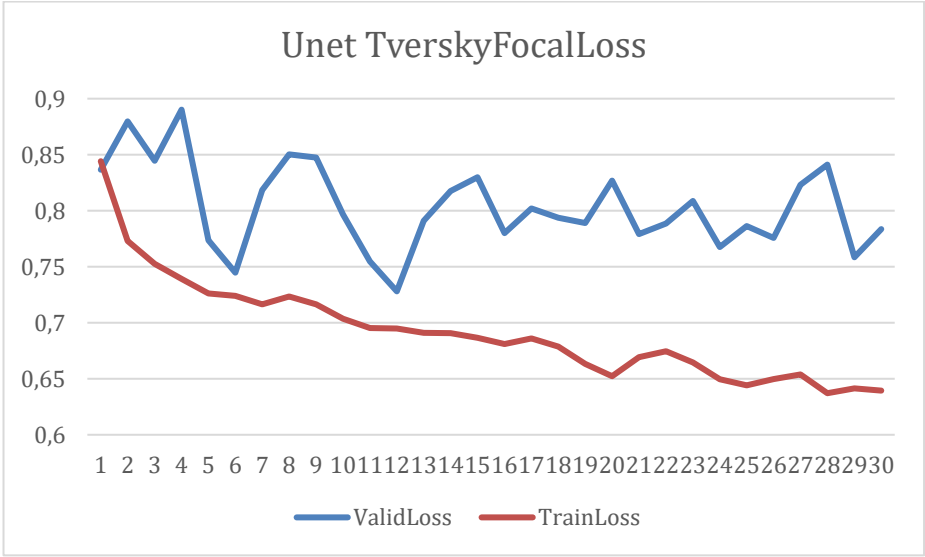

Рисунок 18 - Тренировочные кривые для U-Net, TverskyFocalLoss

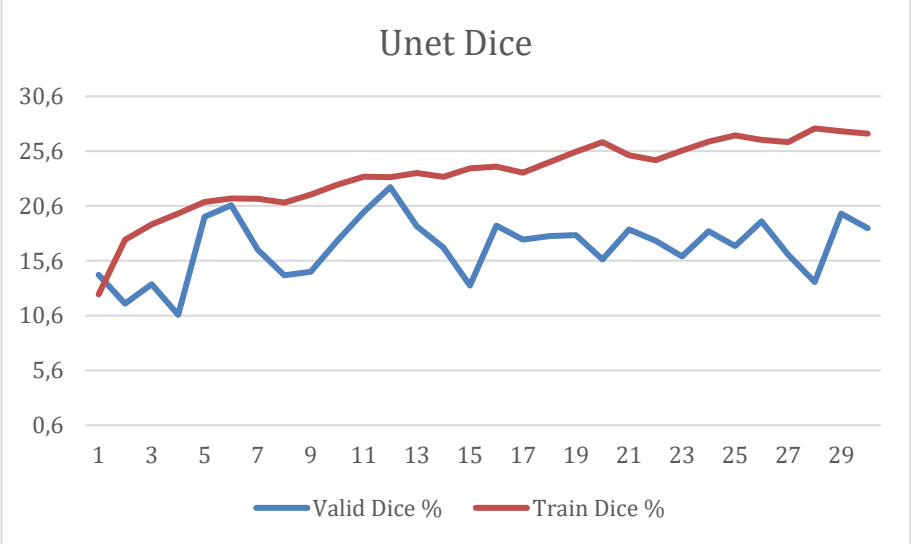

Рисунок 19 - Тренировочные кривые для U-Net, TverskyFocalLoss, коэфициент Дайса

Таблица 1 - Сравнение результатов для U-Net с кодировщиком efficient-net-b5

Loss функция	DiceBCE	DiceBCE	DiceBCE	TverskyFocalLoss
Коэффициенты функций потерь	1	0.1	0.01	a = 0.7, b = 0.3, g = 1.33
Значение функции потерь на валидационной выборке	0.103	0.784	0.825	0.783
Коэфициент Дайса на валидационной выборке	0.071	0.152	0.171	0.186
% площади, занимаемой прожилками	0.99	0.99	0.99	0.99
% площади, занимаемой прожилками предсказанный	1.58	0.79	1.15	0.76
Абсолютная ошибка, %	0.59	0.2	0.16	0.23
Относительная ошибка, %	59	20	16	23

Стоит отметить, что даже при меньшем значении коэфициента Дайса, модели получают меньшую абсолютную ошибку. Это означает, что даже если прожилку определили не на том месте, то это все равно лучше, чем не определить ее вовсе.

В ходе сравнения функций потерь было выяснено, что DiceBCELoss с маленьким коэфициентом BCE, лучше всего справляется с задачей, также неплохо себя показал и TverskyFocalLoss. Как и предполагалось, стандартная бинарная кросс-энтропия получила худшие результаты. Сходимость у DiceBCELoss почти не прослеживается, а TverskyFocalLoss заметно пытался ее уменьшать.

3.3. Attention UNET

Обычная архитектура UNet является довольно старой по мерке машинного обучения, но ее модификации появляются и по сей день. Одна из таких модификаций это Attention UNet, которая использует механизм внимания. Условия тестов совпадают с обычной UNet. Результаты приведены на рисунках 20-31 и в таблице 2.

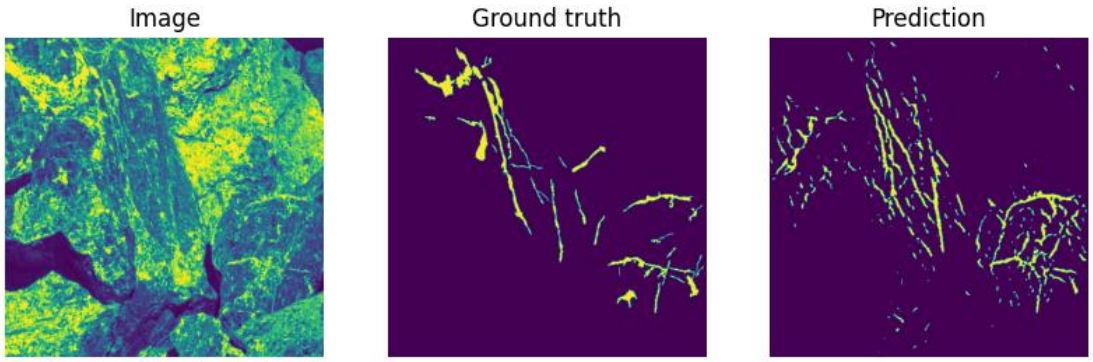

Рисунок 20 - Attention U-Net с DiceBCELoss, коэфициент BCE=1

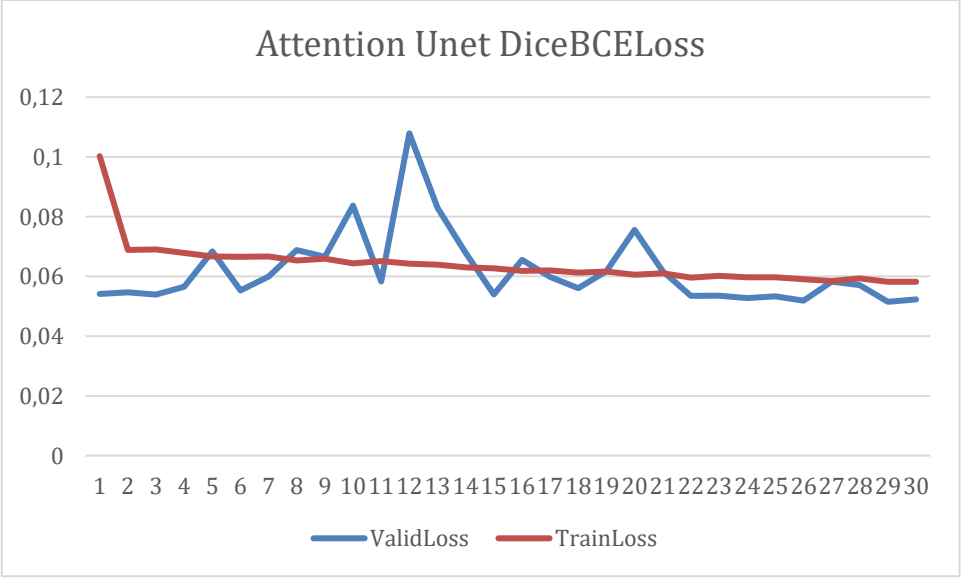

Рисунок 21 - Тренировочные кривые для Attention U-Net, BCE=1, DiceBCELoss

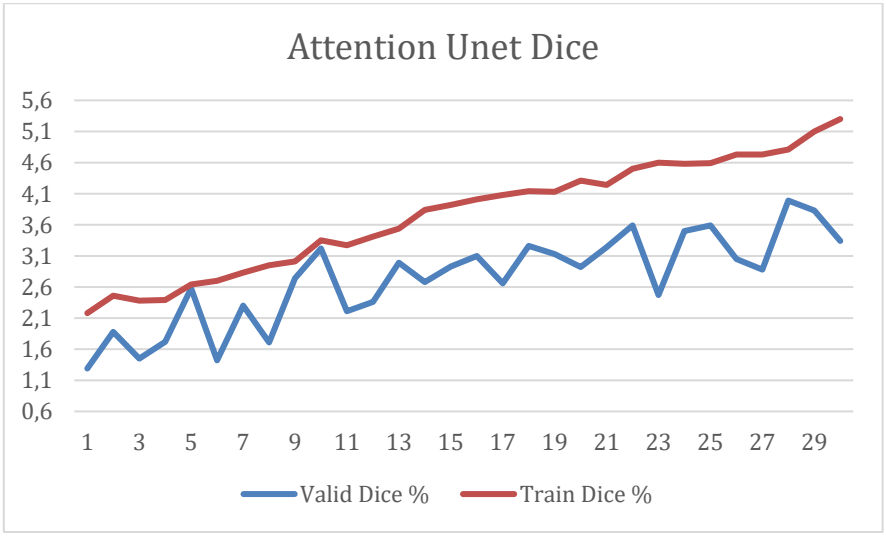

Рисунок 22 - Тренировочные кривые для Attention U-Net, BCE=1, коэфициент Дайса

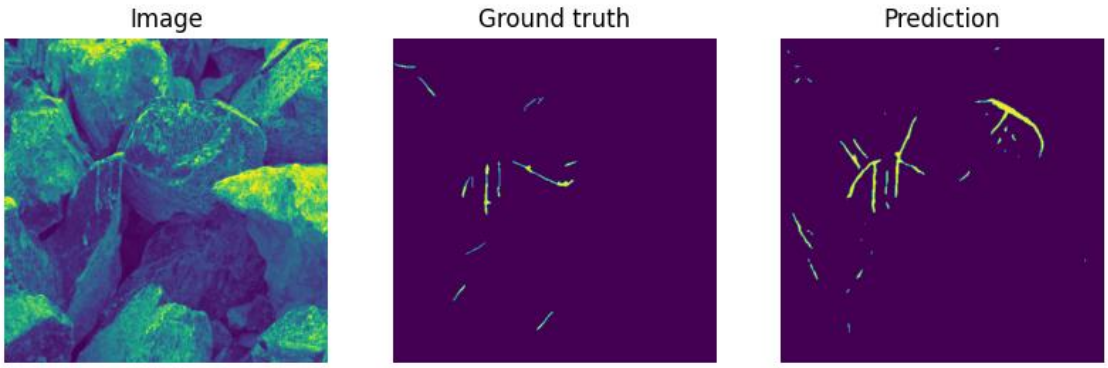

Рисунок 23 - Attention U-Net с DiceBCELoss, коэфициент BCE=0.1

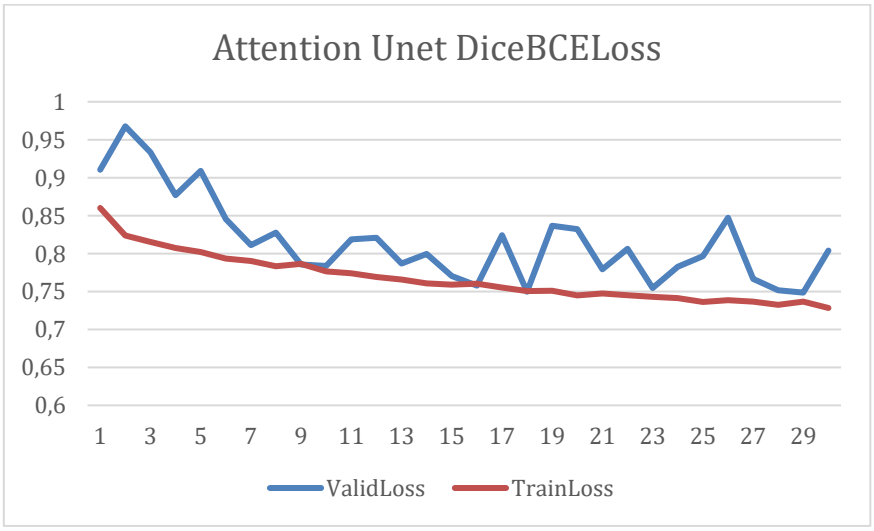

Рисунок 24 - Тренировочные кривые для Attention U-Net, BCE=0.1, DiceBCELoss

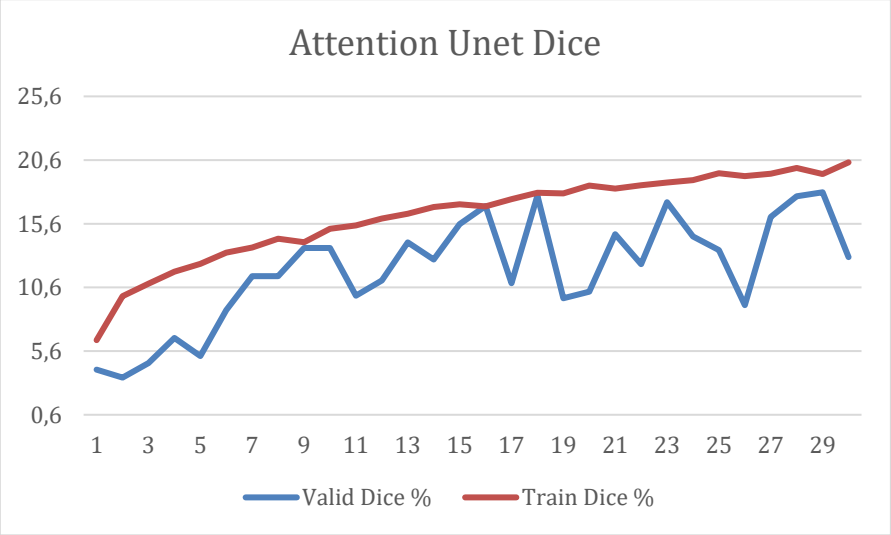

Рисунок 24 - Тренировочные кривые для Attention U-Net, BCE=0.1, коэфициент Дайса

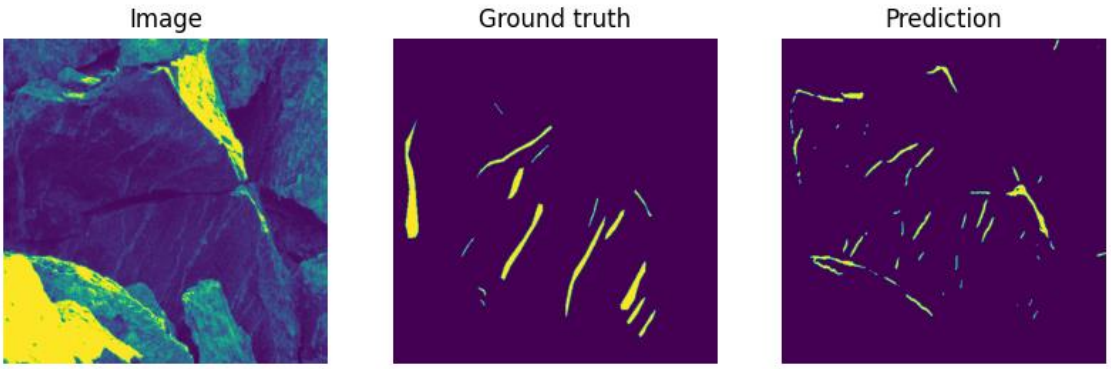

Рисунок 26 - Attention U-Net с DiceBCELoss, коэфициент BCE=0.01

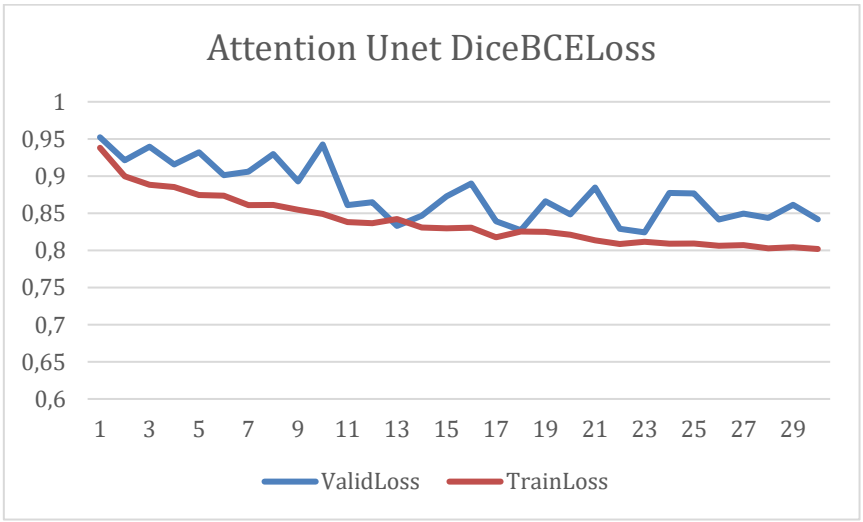

Рисунок 27 - Тренировочные кривые для Attention U-Net, BCE=0.01, DiceBCELoss

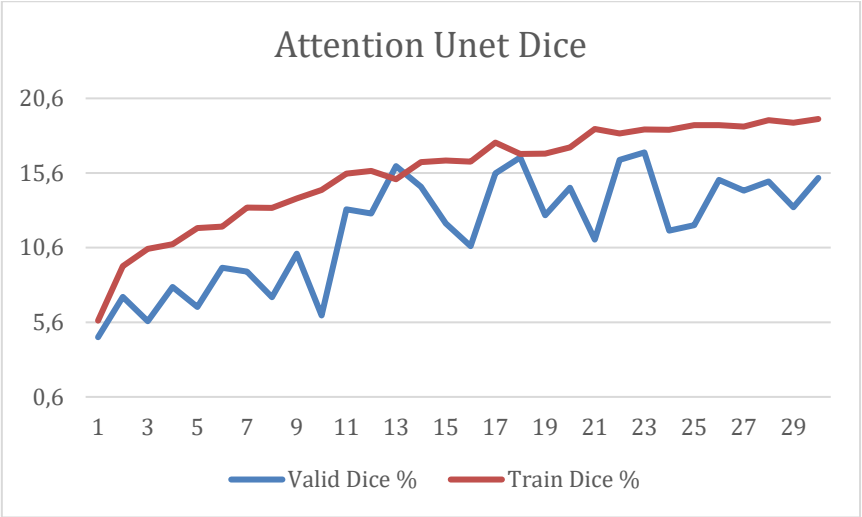

Рисунок 28 - Тренировочные кривые для Attention U-Net, BCE=0.01, коэфициент Дайса

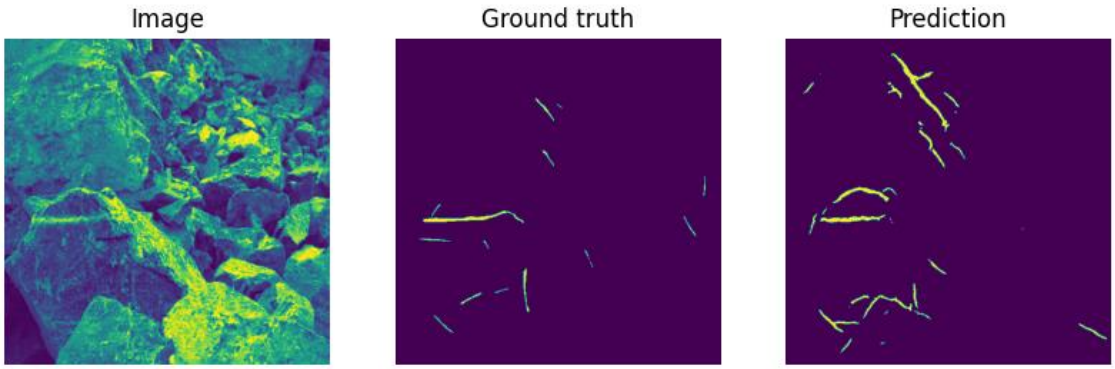

Рисунок 29 - Attention U-Net с TverskyFocalLoss

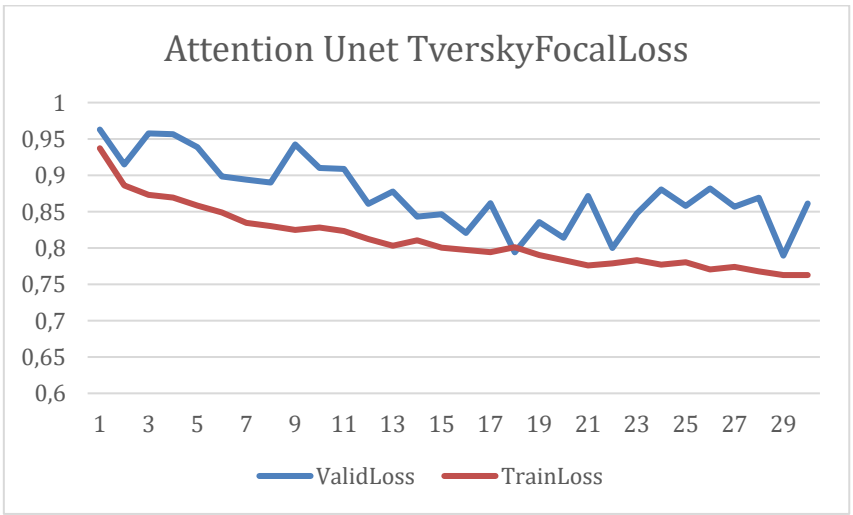

Рисунок 30 - Тренировочные кривые для Attention U-Net, TverskyFocalLoss

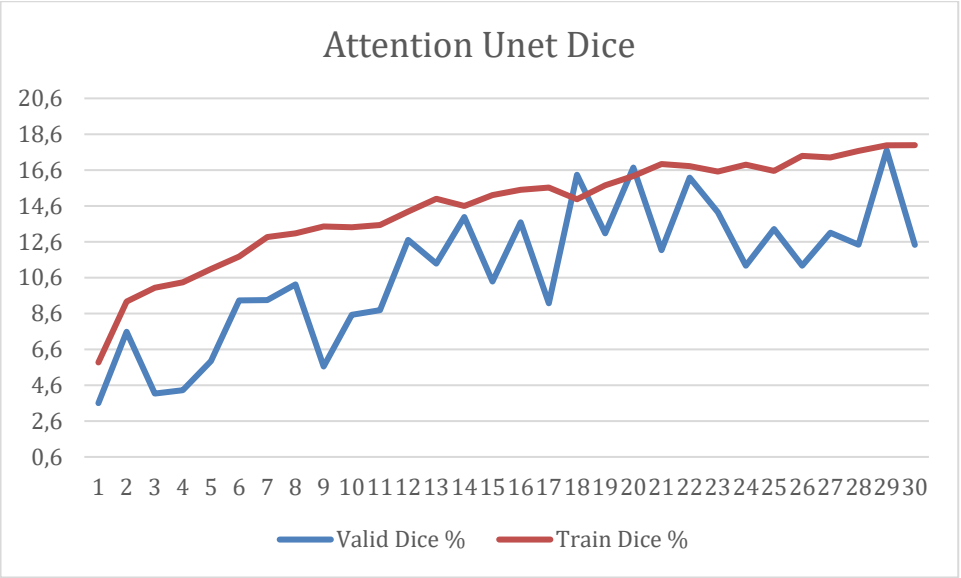

Рисунок 31 - Тренировочные кривые для Attention U-Net, TverskyFocalLoss, коэфициент Дайса

Таблица 2. Сравнение результатов для Attention U-Net

Loss функция	DiceBCE	DiceBCE	DiceBCE	TverskyFocalLoss
Коэффициенты функций потерь	1	0.1	0.01	a = 0.7, b = 0.3, g = 1.33
Значение функции потерь на валидационной выборке	0.052	0.804	0.841	0.861
Коэфициент Дайса на валидационной выборке	0.033	0.129	0.152	0.124
% площади, занимаемой прожилками	0.99	0.99	0.99	0.99
% площади, занимаемой прожилками предсказанный	2.76	2.08	1.53	1.11
Абсолютная ошибка, %	1.77	1.09	0.54	0.12
Относительная ошибка, %	178	110	54	12

В ходе сравнения результатов Attention Unet было выяснено, что TverskyFocalLoss дает наименьшую относительную ошибку и лучше всего

справляется с задачей. Следующей по относительной ошибке оказалась функция потерь DiceBCELoss с коэфициентом при BCE равном 0.01. Сходимость у TverskyFocalLoss все также прослеживается, как и для прошлой модели.

Наибольший коэффициент Серенсена-Дайса равный 0.152 вышел у модели с функцией потерь DiceBCE с коэффициентом при BCE равном 0.01

3.4. Segformer

Более новая архитектура на основе трансформеров. Для обучения были взяты предобученные модели с сайта HuggingFace. В данной архитектуре есть определенная специфика выходных данных. Выходные данные модели на самом деле имеют более низкое разрешение, чем входные. Выходные логиты имеют размер 128x128, тогда как входные — 512x512. Это может отрицательно сказаться на результате сегментации.

Для тестирования были взяты три модели: MIT-B0, MIT-B1 и MIT-B2. Модель обучалась 30 эпох, с оптимизатором Adam и скоростью обучения 0.00006. Результаты на рисунках 32-37 и таблице 3.

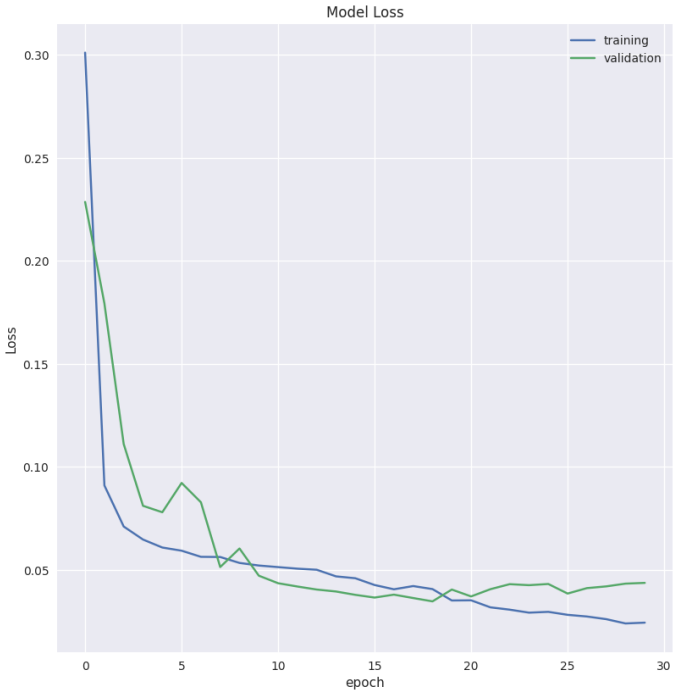

Рисунок 32 - Тренировочные кривые для SegFormer MIT-b0, CrossEntropy

Рисунок 33 - Segformer с backbone MIT-B0

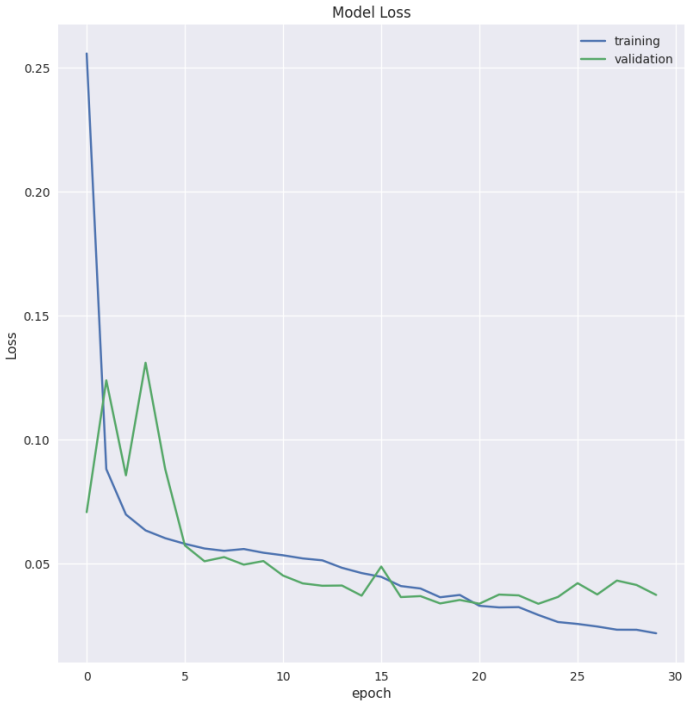

Рисунок 34 - Тренировочные кривые для SegFormer MIT-b1, CrossEntropy

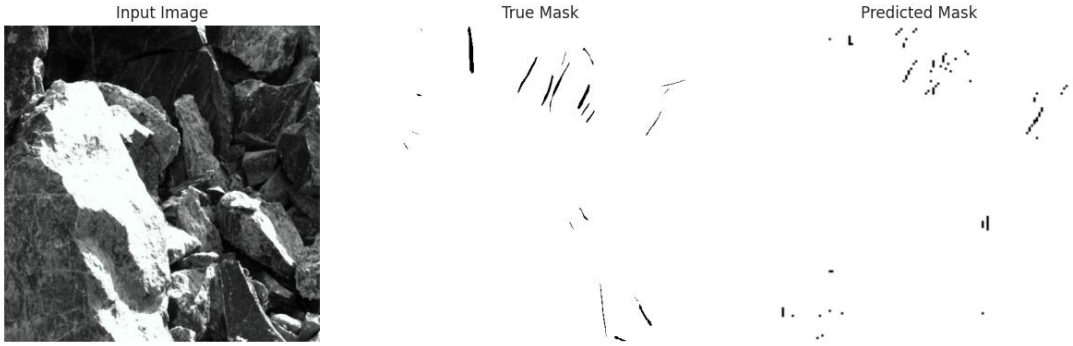

Рисунок 35 - Segformer с backbone MIT-B1

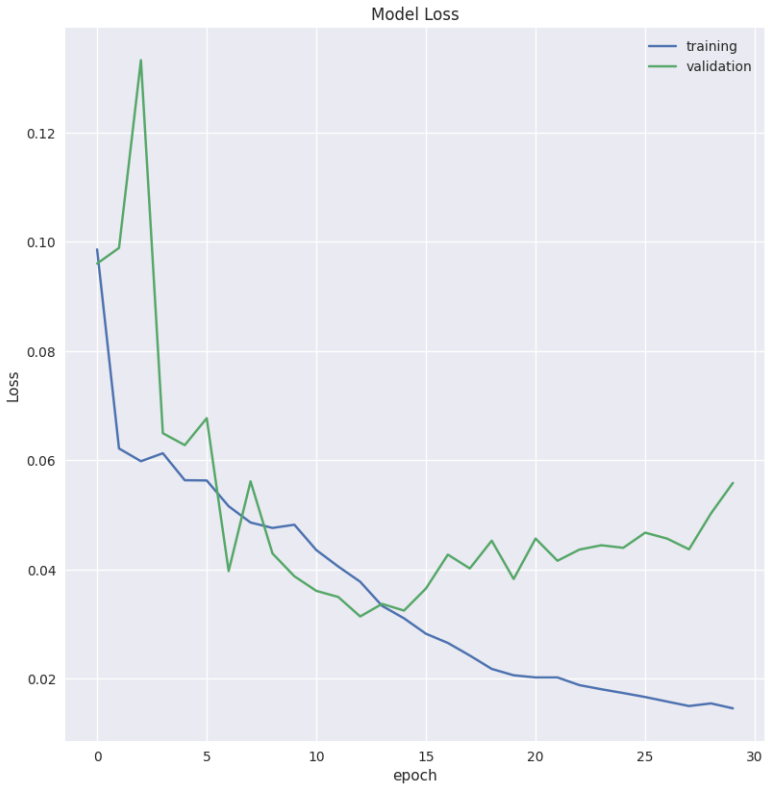

Рисунок 36 - Тренировочные кривые для SegFormer MIT-b2, CrossEntropy

Рисунок 37 - Segformer с backbone MIT-B2

Таблица 3 - Сравнение результатов для U-Net с кодировщиком efficient-net-b5

модель	MIT-B0	MIT-B1	MIT-B2
Вес модели	14 МБ	54 МБ	100 МБ
Значение функции потерь на валидационной выборке	0.043	0.037	0.055

Продолжение таблицы 3

% площади, занимаемой прожилками	0.31	0.31	0.31
% площади, занимаемой прожилками предсказанный	0.049	0.12	0.081
Абсолютная ошибка, %	0.261	0.19	0.229
Относительная ошибка, %	81	61	73

Исходя из таблиц можно сделать вывод, что выбор архитектуры segformer является не самым удачным для данной задачи. Это связано с уменьшенным разрешением выходных логитов, что плохо сказывается на предсказании узких частей и ведет к потере части информации. Еще одни минус — это использование кросс энтропии как функции ошибок без возможности замены на другие. Тем не менее абсолютная ошибка не так велика.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы был проведен анализ предметной области, включающий в себя разбор существующих нейросетевых алгоритмов для семантической сегментации.

В ходе исследования были рассмотрены современные методы семантической сегментации, среди которых особое внимание уделено архитектурам на основе U-Net и SegFormer. Выбор данных архитектур обусловлен их популярностью и относительной новизной, что делает их перспективными для решения задач сегментации. Реализованные модели включали стандартную U-Net, Attention U-Net и модель на основе трансформеров SegFormer. Каждая из этих моделей была обучена и протестирована на специально подготовленном наборе данных, содержащем изображения прожилок в камнях.

В процессе проведения тестирования нейросетевых алгоритмов на наборе данных были получены приемлемые результаты с точки зрения метрик и площади прожилок.

Лучший результат показала модель Attention U-Net с функцией потерь TverskyFocalLoss с относительной ошибкой 12%. Стандартная U-Net показала худший результат среди всех моделей. Модели SegFormer показали средний результат, однако подтвердили свою пригодность для решения задачи сегментации.

Следует отметить, что датасет имел фотографии камней, сделанные на определенном расстоянии. Построенные модели вряд ли будут работать хорошо в других погодных условиях. Для подобных случаев эти результаты должны быть исследованы дополнительно. Несмотря на указанные ограничения, полученные результаты удовлетворяют практическим требованиям. Полученные модели могут быть использованы как средство первичной разметки новых семплов для новой версии датасета.

Перспективы развития проекта включают расширение набора данных за счет добавления изображений, снятых в различных условиях. Это позволит улучшить общую производительность моделей и их способность к обобщению. Также возможно интеграция разработанных алгоритмов в системы автоматического анализа и обработки изображений, что может значительно упростить и ускорить процесс анализа геологических образцов в реальных условиях.

Таким образом, проведенное исследование продемонстрировало высокую эффективность использования нейросетевых алгоритмов для задач семантической сегментации объектов типа прожилки. Достигнутые результаты подтверждают возможность их практического применения и открывают перспективы для дальнейшего развития и усовершенствования. Внедрение предложенных решений способно существенно улучшить качество и скорость обработки данных в геологических исследованиях и смежных областях, что является важным шагом на пути к созданию универсальных и высокоточных систем автоматического анализа изображений.

Полученные результаты имеют потенциал для применения не только в задаче сегментации прожилок, но и в других областях. Например, методы семантической сегментации могут быть адаптированы для анализа медицинских изображений, таких как сегментация опухолей на МРТ и КТ снимках, что способствует улучшению диагностики и планирования лечения.

В строительстве и гражданской инженерии методы семантической сегментации могут применяться для автоматического анализа и мониторинга состояния зданий и инфраструктуры, например, для выявления трещин и дефектов на поверхностях конструкций. В области охраны окружающей среды разработанные алгоритмы могут помочь в мониторинге и оценке состояния лесов, водоемов и других природных объектов, что способствует более эффективному управлению природными ресурсами.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Cai, C. Asbestos Detection with Fluorescence Microscopy Images and Deep Learning / C. Cai, T. Nishimura, J. Hwang, X. Hu, A. Kuroda // Sensors. – 2021 - Vol. 21. № 13. – P. 4582.

2. Zyuzin, V. Automatic Asbestos Control Using Deep Learning Based Computer Vision System / V Zyuzin, M. Ronkin, S. Porshnev, A. Kalmykov // Applied Sciences. – 2021 - Vol. 11. № 22. – P. 10532.

3. Ronneberger O. U-Net: Convolutional Networks for Biomedical Image Segmentation / O. Ronneberger, Ph. Fischer, T. Brox // International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI-2015). – 2015. – Vol. 39. – P. 234–241.

4. Chen, L.-C. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. / L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille // arXiv:1606.00915 – 2017.

5. Chen, L.-C. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. / L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam // arXiv:1802.02611 – 2018.

6. Chen, L.-C. Rethinking Atrous Convolution for Semantic Image Segmentation. / L.-C. Chen, G. Papandreou, F. Schroff, H. Adam // arXiv:1706.05587 – 2017.

7. Chen, L.-C. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. / L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille // arXiv:1412.7062 – 2016.

8. Cordts, M. The Cityscapes Dataset for Semantic Urban Scene Understanding. / M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler // arXiv:1604.01685 – 2016.

9. Лукашик Д.В. Анализ современных методов сегментации изображений / Д.В. Лукашик // Экономика и качество систем связи. – 2022. –

№ 2. – С. 57–65. – URL: https://cyberleninka.ru/article/n/analiz-sovremennyh-metodov-segmentatsii-izobrazheniy/viewer (дата обращения: 25.04.2024)

10. Long J. Fully Convolutional Networks for Semantic Segmentation / J. Long, E. Shelhamer, T. Darrell // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2017. – № 4. – Vol. 39. – P. 640–651.

11. Cristina, S. The Attention Mechanism from Scratch: сайт – URL: https://machinelearningmastery.com/the-attention-mechanism-from-scratch/ (дата обращения: 25.04.2024).

12. Rosebrock, A. Intersection over Union (IoU) for object detection: сайт – URL: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/ (дата обращения: 20.04.2024).

13. The Evolution of Deeplab for Semantic Segmentation B. Sahu : сайт – URL: https://towardsdatascience.com/the-evolution-of-deeplab-for-semantic-segmentation-95082b025571 (дата обращения: 20.04.2024).

14. E. Tiu Metrics to Evaluate your Semantic Segmentation Model: сайт – URL: https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2 (дата обращения: 20.04.2024)

15. Albumentations: Efficient Image Augmentation Library for Machine Learning: сайт – URL: https://albumentations.ai (дата обращения: 20.04.2024).

16. Evaluating image segmentation models. : сайт – URL: https://www.jeremyjordan.me/evaluating-image-segmentation-models/ (дата обращения: 22.03.2024).

17. Transformer в картинках : сайт – URL: https://habr.com/ru/articles/486358/ (дата обращения: 22.03.2024)

18. Badrinarayanan, V. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation / V. Badrinarayanan, A. Kendall, R. Cipolla // arXiv:1511.00561 - 2016.

19. He, K. Mask R-CNN. / K. He, G. Gkioxari, P. Dollár, R. Girshick // arXiv:1703.06870 – 2018.

20. Lin, T.-Y. Feature Pyramid Networks for Object Detection. / T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie // arXiv:1612.03144 – 2017.

21. Oktay, O. Attention U-Net: Learning Where to Look for the Pancreas. / O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich // arXiv:1804.03999 – 2018.

22. Ren, S. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. / S. Ren, K. He, R. Girshick, J. Sun // - 2016.

23. Vaswani, A. Attention Is All You Need. / A. Vaswani, N. Shazeer , N. Parmar, J. Uszkoreit, L. Jones // arXiv:1706.03762 – 2023.

24. Xie, E. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. / E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, P. Luo // arXiv:2105.15203 – 2021/

25. Liu, Z. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. / Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei // arXiv:2103.14030 – 2021.

26. Deng, J. A Large-Scale Hierarchical Image Database. // CVPR09 – 2009.

27. Semantic Segmentation | Papers With Code: сайт – URL: https://paperswithcode.com/task/semantic-segmentation (дата обращения: 28.05.2024)

28. Lin, G. RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation. / G. Lin, A. Milan, C. Shen, I. Reid // arXiv:1611.06612 - 2016.

1. Otsu, N. A Threshold Selection Method from Gray-Level Histograms / N. Otsu – Текст : непосредственный. // IEEE Transactions on Systems, Man, and Cybernetics. 1979. Т. 9. № 1. – С. 62-66.

29. Otsu, N. A Threshold Selection Method from Gray-Level Histograms / N. Otsu // IEEE Transactions on Systems, Man, and Cybernetics. - 1979. - Т. 9. № 1. – С. 62-66.

30. Noh, H. Learning Deconvolution Network for Semantic Segmentation / H. Noh, S. Hong, B. Han // arXiv:1505.04366 – 2015.

31. Tan, M. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks / M. Tan, Q.V. Le // arXiv:1905.11946 – 2020.

32. Yesilkaynak, V. B. EfficientSeg: An Efficient Semantic Segmentation Network / V.B. Yesilkaynak, Y.H. Sahin, G. Unal // arXiv:2009.06469 - 2020. EfficientSeg.

33. The PASCAL Visual Object Classes Homepage: сайт – URL: http://host.robots.ox.ac.uk/pascal/VOC/ (дата обращения: 29.05.2024)

34. Lin, T.-Y. Microsoft COCO: Common Objects in Context. / T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick // arXiv:1405.0312 - 2015. Microsoft COCO.

35. Zhou, Z. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. / Z. Zhou, R. Siddiquee, N. Tajbakhsh, J. Liang // arXiv:1807.10165 - 2018. UNet++.

36. Mcculloch, W. – Pitts, W. A Logical Calculus of Ideas Immanent in Nervous Activity.// Bulletin of Mathematical Biophysics – 1943 - 5, s. 127–147.

37. Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. // Psychological Review. – 1958 - s. 65–386.

38. Widrow, B. Adaline: Smarter than sweet // Stanford today - 1963

39. Minsky, M. – Papert, S. A. Perceptrons: An Introduction to Computational Geometry. // The MIT Press – 2017 - ISBN 0262534770.

40. torch.nn // PyTorch documentation: сайт – URL: https://pytorch.org/docs/stable/nn.html (дата обращения: 20.05.2024)

41. Salehi S.S.M. Tversky loss function for image segmentation using 3D fully convolutional deep networks / S.S.M. Salehi, D. Erdogmus, A. Gholipour // Machine Learning in Medical Imaging. Proceedings of 8th International Workshop MLMI 2017. – 2017. – P. 379–387.

42. Segmentation_models.pytorch // GitHub. – URL: https://github.com/qubvel/segmentation_models.pytorch/tree/master (дата обращения: 20.05.2024)

43. Werbos, P. J. Backpropagation through time: what it does and how to do it. // Proceedings of the IEEE – 1990 - 78, 10, s. 1550–1560.

44. Rumelhart, D. E. Learning representations by back-propagating errors. / D. E. Rumelhart, G. E. Hinton, R. J. Williams // Nature 323 – 1986 - 533–536

45. Lecun, Y. Gradient-based learning applied to document recognition. / Y. Lecun, L. Bottou, Y. Bengio and P. Haffner // Proceedings of the IEEE. – 1998 - 86, 11, s. 2278–2324.