Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное
учреждение высшего образования
Национальный исследовательский Нижегородский государственный
университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Отчет по лабораторной работе

«U-Net»

Выполнил:
студент группы 3823Б1Пмоп2
Золкин И.А.

Проверил:
Доцент каф. ТУиДС
Смирнов Л. А.

Нижний Новгород
2025

Содержание

Введение

В области компьютерного зрения задача сегментации является одной из фундаментальных и практически значимых. Её решения находят применение в медицинской диагностике (анализ снимков МРТ, КТ), автономном вождении (выделение дорог, пешеходов), дистанционном зондировании Земли и многих других областях. Традиционные алгоритмы обработки изображений часто не справляются с вариативностью и сложностью реальных данных, что обуславливает необходимость использования глубокого обучения.

Основная сложность в сегментации заключается в необходимости совмещать два, казалось бы, противоречивых требования:

1) Общее понимание изображения на глобальном уровне для корректной классификации объектов.

2) Точное локализованное соответствие на уровне пикселей для формирования чётких границ объектов.

Стандартные свёрточные сети, успешные в классификации, теряют пространственную детализацию из-за последовательных операций пулинга.

В 2015 году Олаф Роннебергер, Филлип Фишер и Томас Брокс предложили архитектуру U-Net, которая стала прорывным подходом, эффективно решающим указанную проблему.

Благодаря своей U-образной структуре архитектура стала способная комбинировать детализированную информацию из ранних слоев с высокоуровневыми признаками из глубоких слоёв, что обеспечило точное позиционирование границ объектов.

Целью данной работы является теоретическое и практическое исследование архитектуры нейронной сети U-Net.

Для достижения цели требуется выполнить следующие задачи:

1) Изучить принципы работы, детали архитектуры и математический аппарат, лежащий в основе UNet.
2) Реализовать модель UNet на фреймворке глубоко обучения PyTorch, PyTorch_Lightning, настроить логгирование с использованием ML-Flow.
3) Обучить модель на специализированном наборе для сегментации.
4) Привести оценку результатов сегментации с использованием стандартных метрик, таких как Dice Coefficient, BCEWithLogitsLoss.
5) Проанализировать влияние ключевых компонентов архитектуры на конечное качество модели.

Описание датасета

Для обучения модели была использована коллекция данных Human Segmentation Dataset [1], предоставленная на платформе Kaggle. Данный датасет предназначен для задачи бинарной семантической сегментации с фокусом на выделение фигуры человека из фона.

Общая характеристика:

Датасет содержит 2667 изображений в формате PNG с соответствующими бинарными масками сегментации и дополнительными коллажами. Изображения представлены в исходном высоком разрешении с вариативными размерами, что обеспечивает разнообразие данных.

Структура данных:

В корневой директории датасета расположен файл df.csv и директории images, masks и collage.

В файле df.csv находится описание путей файлов датасета. Файл содержит 4 столбца: номер, пути до изображений (”images”), пути масок (”masks”), пути коллажей (”collages”), изображения в формате png.

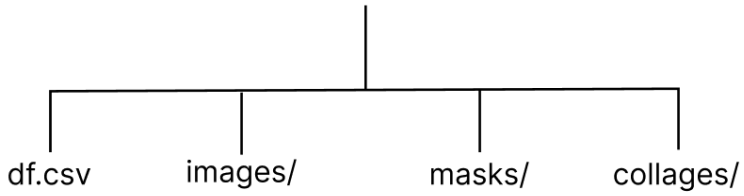

Рис. 1. Организация директории датасета.

	A	B	C	D	E	F
1		images	masks	collages		
2	0	images/ds10_	masks/ds10_	collage/ds10_pexels-photo-687782.jpg		
3	1	images/ds10_	masks/ds10_	collage/ds10_pexels-photo-835971.jpg		
4	2	images/ds10_	masks/ds10_	collage/ds10_pexels-photo-850708.jpg		
5	3	images/ds10_	masks/ds10_	collage/ds10_pexels-photo-864937.jpg		
6	4	images/ds10_	masks/ds10_	collage/ds10_pexels-photo-865908.jpg		

Рис. 2. содержание df.csv

Статистические особенности:

1. Изображения охватывают различные условия съёмки (освещение, ракурс, фон).
2. Люди представлены в разных позах и масштабах.

Визуальный анализ показывает примерный баланс между площадью объекта (человека) и фона, что снижает риск смещения модели при обучении.

Рис. 3. Пример изображения из датасета и соответствующей ему маски.

Архитектура сети

Unet [2] - это сверточная нейронная сеть (Convolutional Neural Network, CNN) с U-образной симметрической архитектурой, специально разработанная для задач семантической сегментации биомедицинских изображений. UNet показывает хорошие результаты даже при работе с малым количеством данных.

Ключевая особенность UNet - способность сочетать контекстуальную информацию (что изображено) с точной пространственной локализацией (где именно находятся объекты), что является центральной проблемой в сегментации.

Архитектура UNet состоит из двух симметричных путей и моста:

1. Encoder - предназначен для извлечения признаков и контекстной информации из изображения.

   Состоит из повторяющихся блоков, каждый из которых включает две свертки 3x3 с активацией ReLU и операцию макс-пулинга 2x2 с шагом 2 для понижения пространственной размерности.

   На каждом уровне сжатия количество карт признаков удваивается, о пространственные размеры уменьшаются вдвое, что позволяет сети изучать иерархические признаки - от простых границ и текстур к сложным объектам и сценам.

2. Decoder - предназначен для точного пространственного восстановления сегментационной маски.

   Состоит из симметричных блоков, каждый из которых включает операцию транспонированной свертки для увеличения пространственных размеров, конкатенацию с соответствующими картами признаков из пути сжатия через skip-connections и две свертки 3x3 с активацией ReLU.

   На каждом уровне расширения количество карт признаков уменьшается вдвое, а пространственные размеры увеличиваются, что позволяет постепенно восстанавливать детализированную сегментационную маску.

3. Bottleneck - самый нижний слой архитектуры, соединяющий путь сжатия и путь расширения, содержит наиболее абстрактные и высокоуровневые признаки.

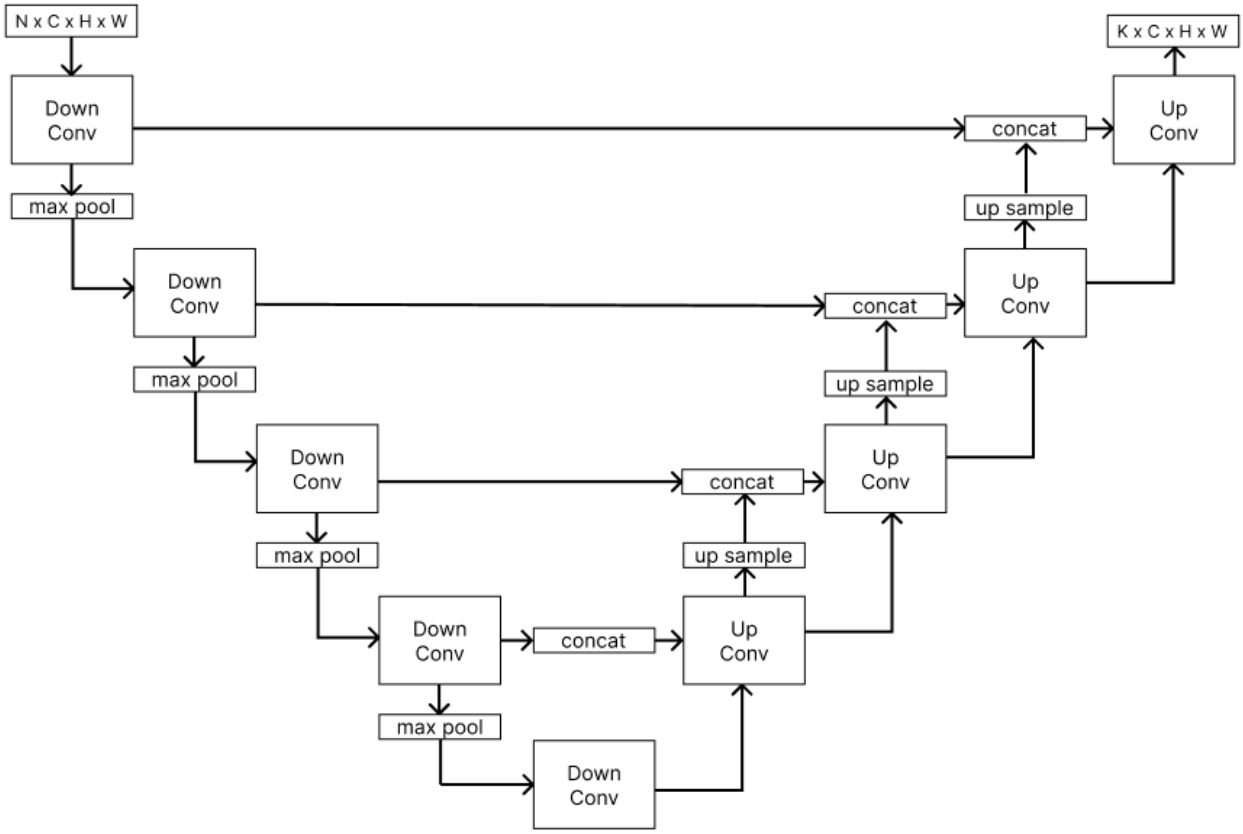

Рис. 4. Реализованная архитектура Unet.

Обработка данных

Перед обучением модели все изображения и соответствующие маски сегментации подвергались стандартизированной обработке, направленной на приведение данных к единому формату, совместимому с архитектурой нейронной сети. Процесс обработки включает следующие этапы:

Изменение размера:

Исходные изображения и маски имеют переменные размеры, что недопустимо для батчевой обработки в нейронных сетях. Для унификации все пары “изображение-маска” приводятся к фиксированному размеру 256x256 пикселей.

Преобразование цветовых пространств:

Исходные изображения преобразуются в трёхканальный формат RGB. Это обеспечивает совместимость с первым свёрточным слоем модели, ожидающим 3 входных канала. Маски сегментации преобразуются в одноканальный формат (grayscale), где значения пикселей представляют бинарные метки:

0 - фон,

255 - целевой объект (человек).

Тензорное преобразование:

Все данные преобразуются в тензоры PyTorch,

Обучение модели

В качестве функции потерь при обучении модели сегментации использовалась Binary Cross-Entropy with Logits Loss (nn.BCEWithLogitsLoss).

Для ранней остановки обучения и мониторинга использовалась метрика Dice=(2 * TP) / (2 * TP + FP + FN).

В процессе обучения использовался оптимизатор Adam (Adaptive Moment Estimation) и планировщик скорости обучения ReduceLROnPlateau в режиме ‘min’, который уменьшает Learning rate вдвое по прошествии 5 эпох. Мониторинг происходит по val_loss.

Adam относится к классу адаптивных оптимизаторов первого порядка, который вычисляет индивидуальные адаптивные скорости обучения для каждого параметра модели. Он вычисляет скользящие средние как градиентов, так и квадратов градиентов, что позволяет быстро сходится в начале обучения, поддерживать стабильное обучения на поздних этапах, автоматически адаптировать скорость обучения для каждого параметра и эффективно работать с зашумленными градиентами.

Условия при обучении модели:

1) Максимальное количество эпох - 100.
2) Размер батча - 4.
3) Начальный learning rate - 1e-5.
4) Использовался градиентный клиппинг с выставленным значением 1.
5) Осуществлялась ранняя остановка по прошествии 15 эпох без улучшения метрики Dice. Минимальная погрешность при сравнении равна 1e-4.

Обучение происходило на GPU: NVIDIA GeForce RTX 3060 Laptop GPU (6 GB).

Метрики

Для оценки качества модели сегментации были выбраны следующие метрики [3]:

Функция потерь: Binary Cross-Entropy (BCE) with Logits Loss - данная функция потерь сочетает операцию сигмойды и бинарную кросс-энтропию в одной стабильной с вычислительной точки зрения функции. Она измеряет расхождения между предсказанными вероятностями и истинными бинарными масками. Служит хорошей дифференцируемой функцией потерь для градиентного спуска и является стандартным выбором для задача бинарной сегментации. Она хорошо работает когда классы сбалансированы.

Dice = (2 * |XY|) / (|X| + |Y|), где X - предсказанная маска, Y - истинная маска. Основная метрика для задач сегментации. Непосредственно измеряет площадь перекрытия между предсказанием и истиной.

Выбрана в качестве основной метрики для сохранения лучших моделей из-за своей устойчивости к несбалансированным данным (когда фон доминирует над объектом).

Также для вычисления Dice на уровне эпохи используются бинарные статистические показатели:

True Positives (TP) - правильные положительные предсказания,

False Positives (FP) - ложные положительные предсказания,

True Negatives (TN) - правильные отрицательные предсказания,

False Negatives (FN) - ложные отрицательные предсказания.

Результаты

Обучение завершилось на 44-й эпохе из-за срабатывания раннего выхода по метрике Dice. Лучший результат по валидационной выборке был на 29й эпохе. В процессе обучения был заметен рост метрики dice к единице для тренировочной выборки, что говорит о правильном процессе обучения модели. По этой же причине заметно уменьшение train_loss.

В тоже время метрики валидационного набора улучшались до 29-й эпохи включительно, а дальше улучшение прекращалось - loss начал ухудшаться.

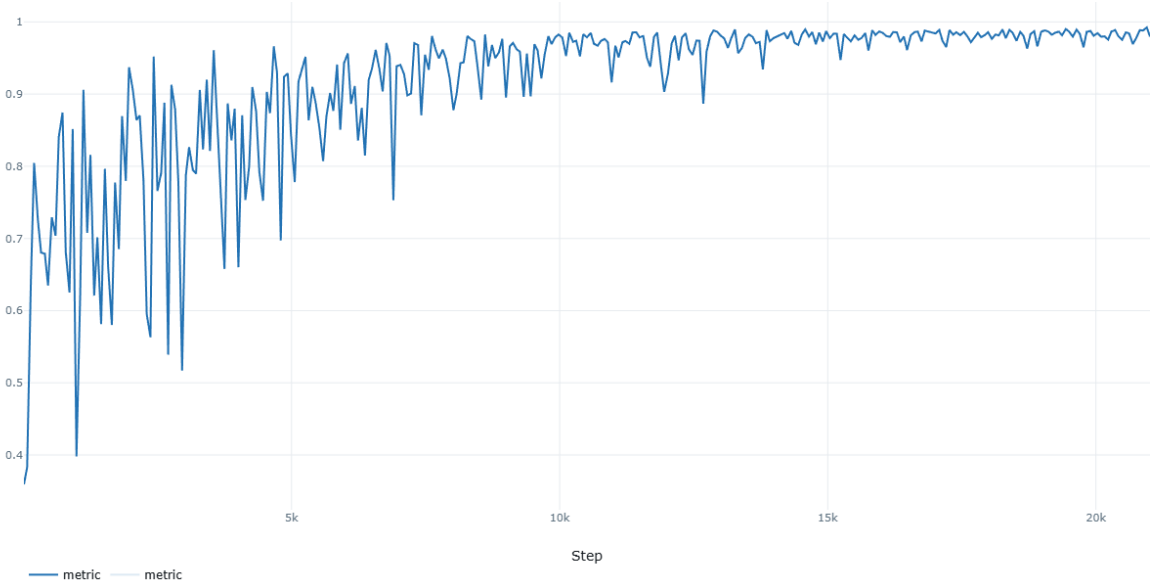

Рис. 5. train_dice - метрика dice пошагово для обучающей выборки.

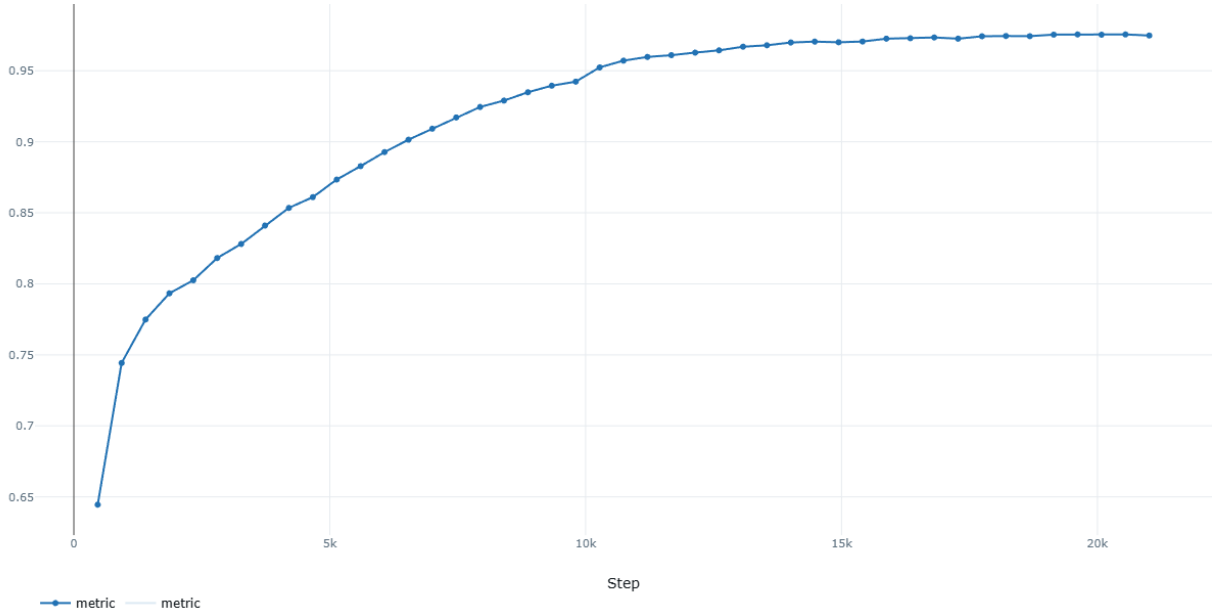

Рис. 6. train_dice_epoch - метрика Dice для каждой эпохи обучающей выборки.

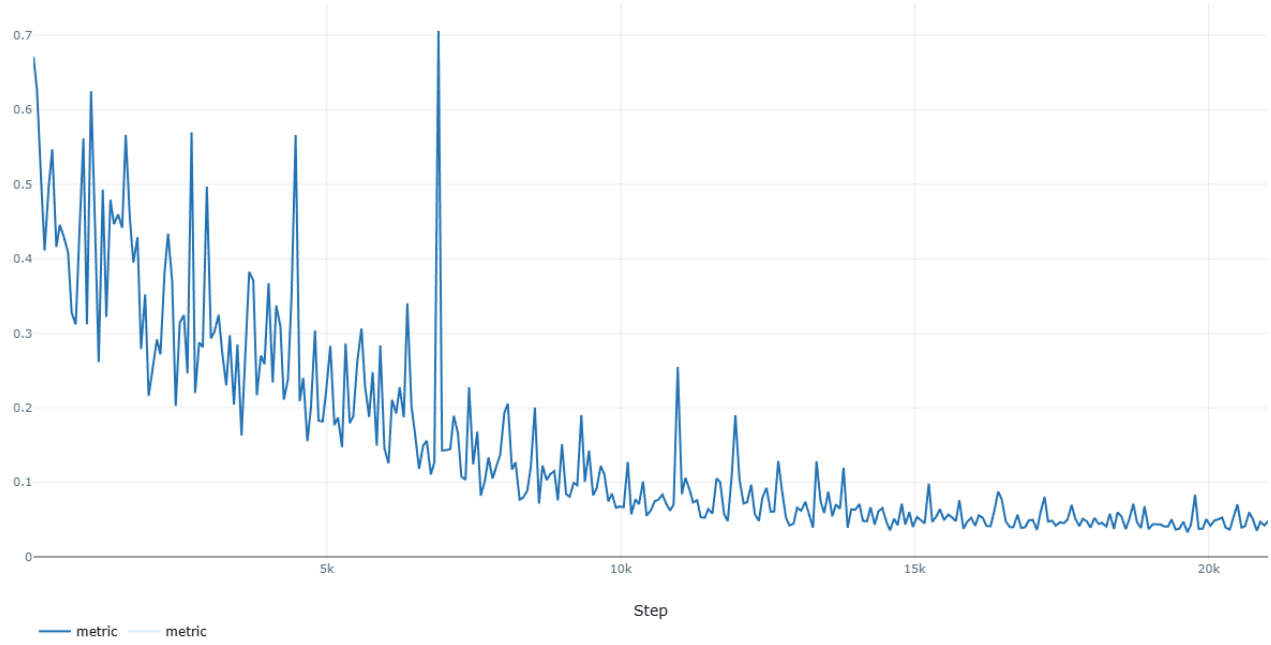

Рис. 7. train_loss для обучающей выборки.

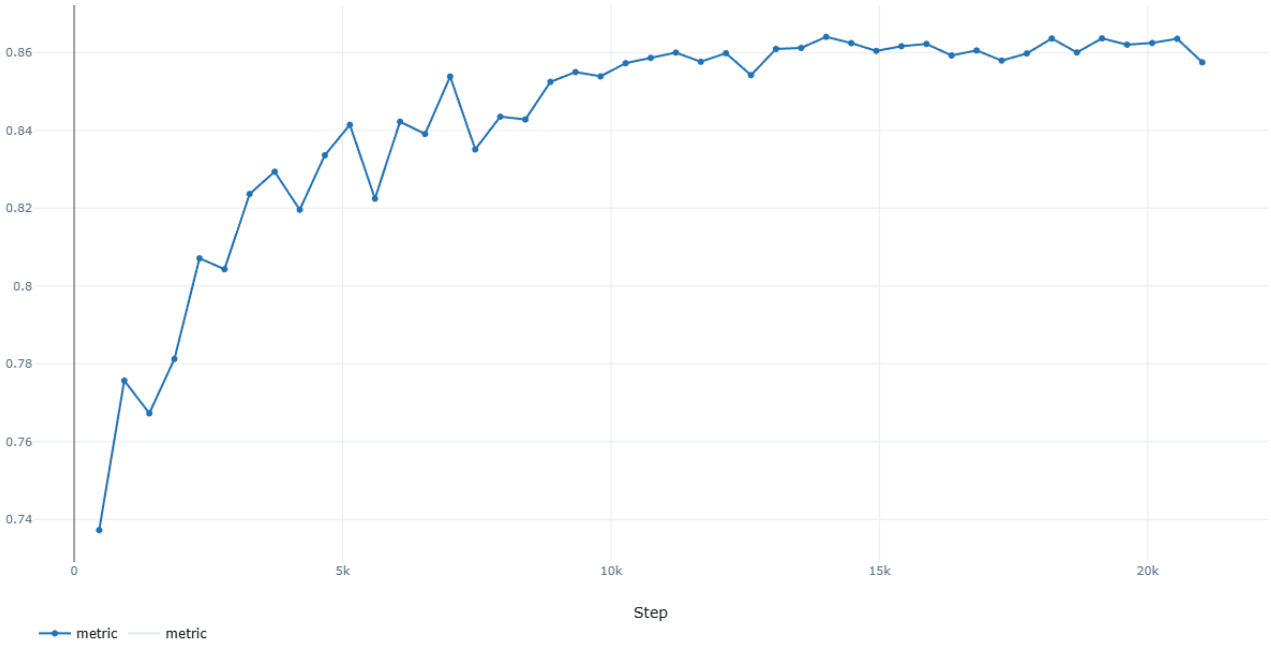

Рис. 8. val_dice_epoch - метрика Dice для каждой эпохи валидационной выборки.

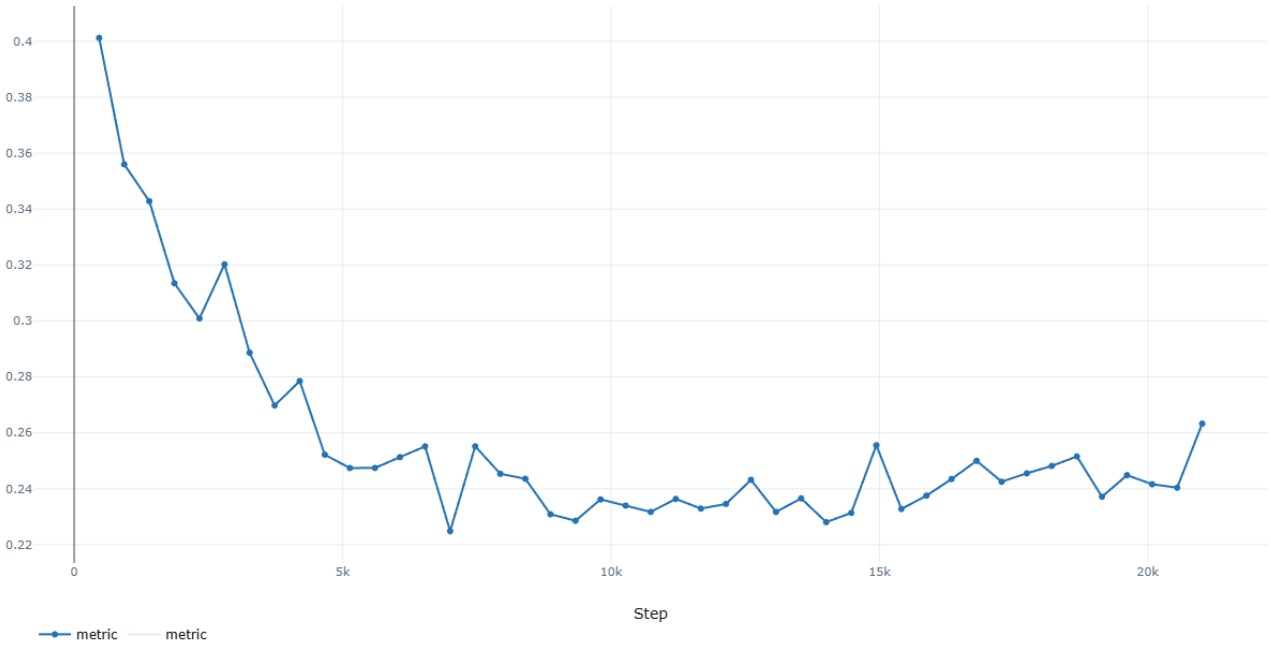

Рис. 9. val_loss валидационной выборки.

Проверка модели на тестовой выборке показала такие результаты:

Dice - 0.8556106686592102

Loss - 0.2595876157283783

Визуализация работы модели на случайном изображении из датасета:

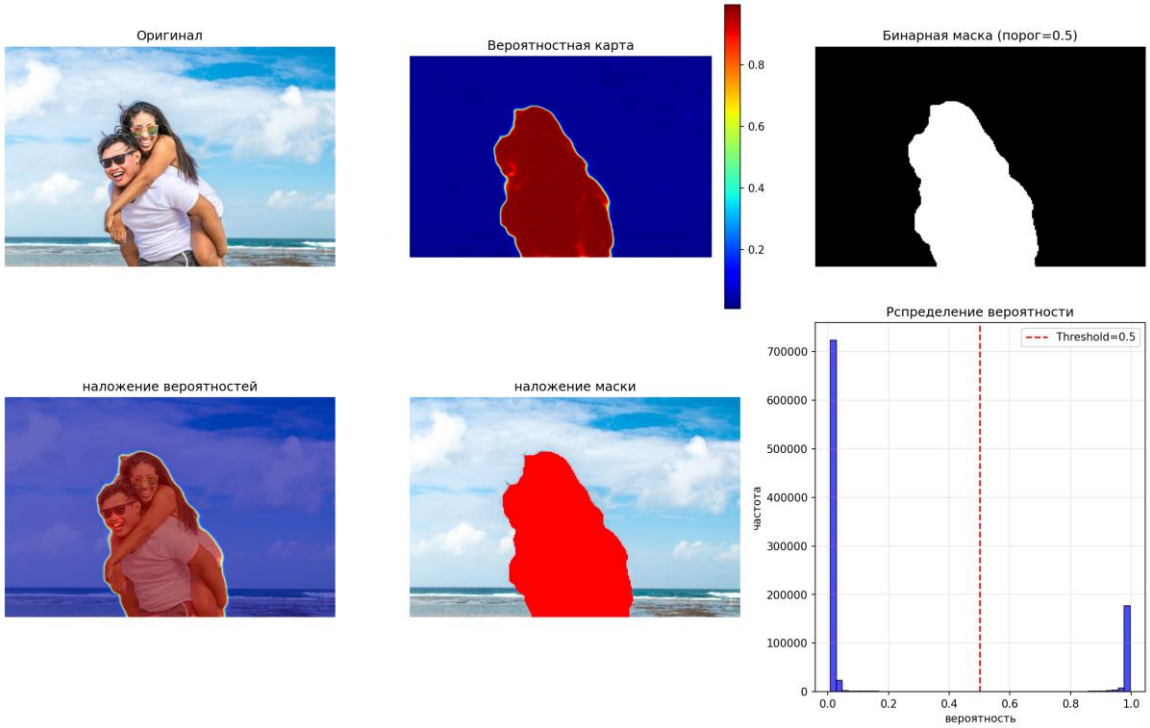

Рис. 10. - работа сети на случайном изображении из датасета.

Визуализация работы модели на случайном изображении не из датасета (из открытого доступа):

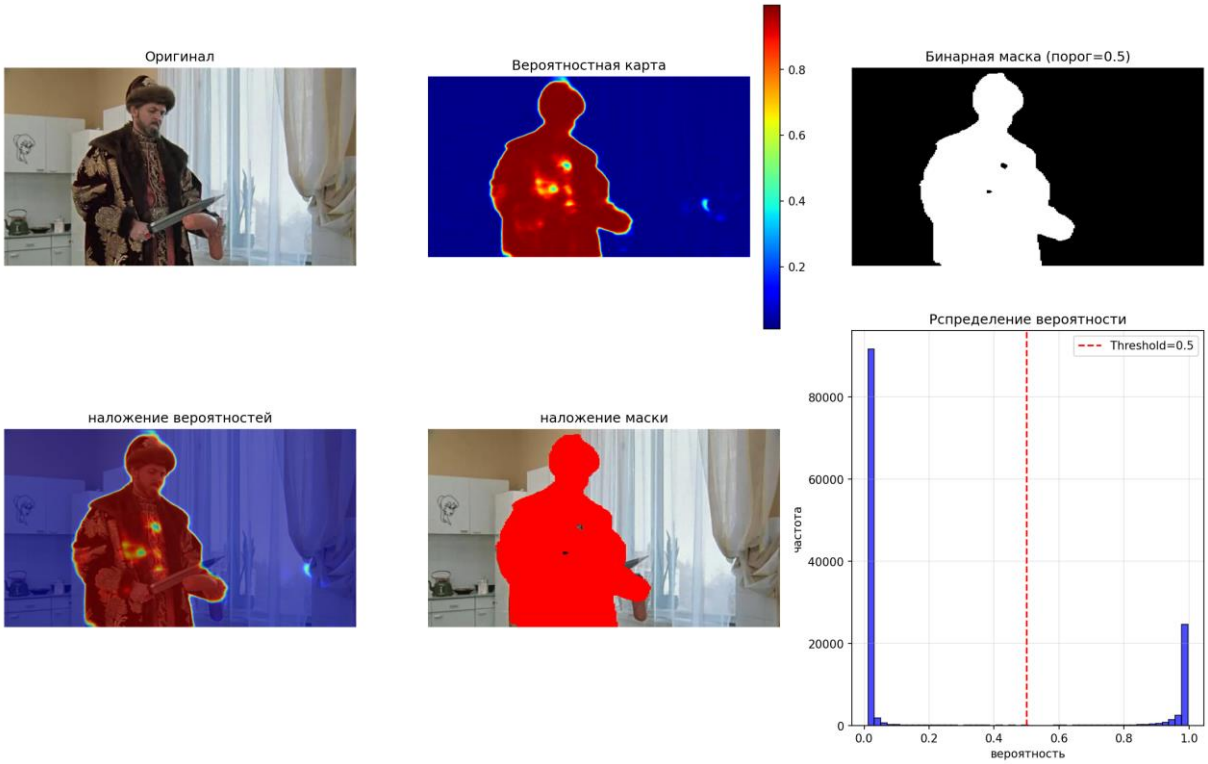

Рис. 11. - работа сети на случайном изображении из открытых источников (не присутствует в датасете)

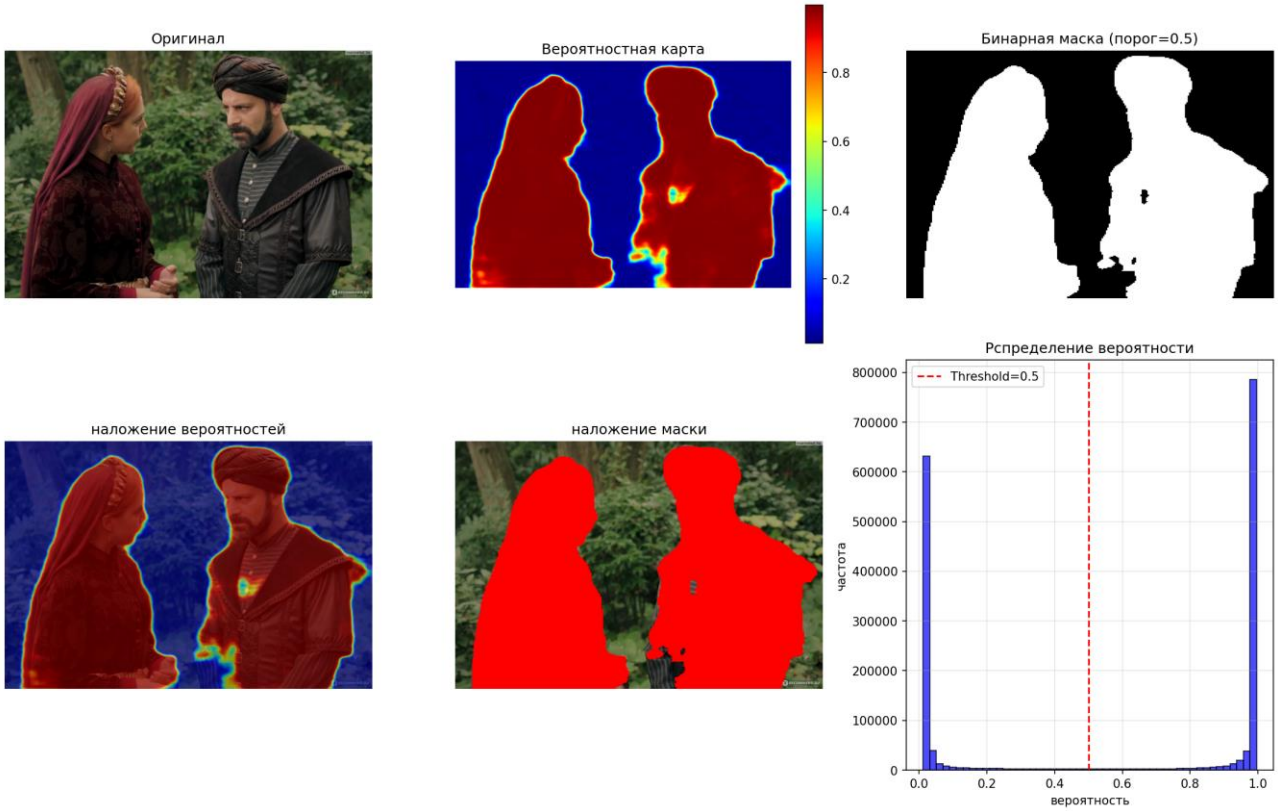

Рис. 12. - работа сети на случайном изображении из открытых источников (не присутствует в датасете)

Выводы

Проведённое исследование позволило сделать следующие выводы относительно применения архитектуры U-Net для бинарной семантической сегментации:

Подтверждение эффективности базовой архитектуры:

Реализованная модель U-Net, несмотря на свою относительно простую структуру, показала высокую практическую эффективность. Ключевой механизм работы - skip-connections, обеспечивающий объединение карт признаков из энкодера и декодера, - успешно решает фундаментальную проблему сегментации: совмещение контекстуальной информации с точным позиционированием границ. Конечная метрика Dice ≈ 0.856 является убедительным подтверждением этого.

Динамика обучения и проблема переобучения:

Анализ графиков обучения (*train_loss, val_loss, val_dice*) выявил характерную динамику. После 29-й эпохи наблюдалось продолжение улучшения метрик на обучающем наборе, в то время как на валидационном наборе качество ухудшалось. Это является классическим признаком начала переобучения, когда модель начинает запоминать специфические особенности тренировочных данных, теряя способность к обобщению. Использованный стратегический приём - ранняя остановка по метрике Dice на валидации - доказал свою целесообразность, позволив остановить обучение, не допустив дальнейшую деградацию модели.

Качественная оценка результатов:

Визуальный анализ результатов сегментации на тестовых данных и, что особенно важно, на внешних изображениях показал, что модель корректно выделяет целевой объект (человека) на разнообразном фоне. Наблюдаемые ошибки, как правило, связаны со сложными случаями: нестандартная поза, низкая контрастность между объектом и фоном.

Перспективы оптимизации:

Полученные результаты создают основу для дальнейшего совершенствования системы. Предполагаемые способы улучшения эффективности модели:

1) Добавление аугментации данных - комбинации изображений.
2) Модификация архитектуры - увеличение глубины сети.
3) Увеличение размера и разнообразия датасета.

Список литературы

[1] Kaggle: Supervisely Filtered Segmentation Person Dataset / Tapakah68. – URL: https://www.kaggle.com/datasets/tapakah68/supervisely-filtered-segmentation-person-dataset.

[2] Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation // Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. – Lecture Notes in Computer Science. – 2015. – Vol. 9351. – P. 234–241.

[3] Taha A.A., Hanbury A. Metrics for Evaluating 3D Medical Image Segmentation: Analysis, Selection, and Tool // BMC Medical Imaging. – 2015. – Vol. 15. – Article 29.