

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

УДК 004.021

**Отчет о программном проекте на тему:
Поиск лесных дорог на спутниковых снимках**

Выполнили:

студент группы БПМИ201
Ахматова Анна Увайсовна

(подпись)

(дата)

студент группы БПИ211
Степанов Артемий Романович

(подпись)

(дата)

Принял руководитель проекта:

Старичков Никита Юрьевич
Внештатный преподаватель (по ГПХ)
Факультета компьютерных наук НИУ ВШЭ

(подпись)

(дата)

Москва 2023

Содержание

Аннотация	3
1 Введение	4
2 Обзор литературы	5
2.1 Fully Convolutional Networks for Semantic Segmentation	5
2.2 U-Net: Convolutional Networks for Biomedical Image Segmentation	6
2.3 Rethinking Atrous Convolution for Semantic Image Segmentation	7
2.4 Automatic Rural Road Centerline Detection and Extraction from Aerial Images for a Forest Fire Decision Support System	9
2.5 Detection of forest roads in Sentinel-2 images using U-Net	12
3 Создание датасета	13
4 Подбор архитектуры нейронной сети	14
4.1 Архитектура ResNet50	14
4.2 Архитектура fcn_resnet50	16
4.3 Архитектура UNet	16
4.4 Архитектура deeplabv3_resnet50	16
4.5 Результаты экспериментов	17
5 ForestRoadsExtractor	19
6 Анализ результатов	21
7 Приложение FOREST NAVIGATOR	22
8 Заключение	23
Список литературы	25

Аннотация

Данный курсовой проект был посвящен созданию модели машинного обучения, распознающей лесные дороги по спутниковым снимкам. Модель - глубокая сверточная нейронная сеть. Модель решает задачу семантической сегментации, то есть по спутниковому снимку получает бинарную маску, каждому пикселю исходного изображения предсказывает его класс: 0, если это фон, и 1, если это дорога.

В ходе проекта разработан алгоритм обработки снимков моделями и перевода растровых бинарных масок, описывающих карты лесных дорог, в векторный формат osm.

Разработанная модель также была встроена в мобильное приложение для ее удобного использования. В приложении пользователь указывает интересующую его местность наведением на интерактивной карте, приложение обрабатывает снимок с помощью нейронной сети и рисует получившуюся бинарную карту слоем поверх снимка.

Ключевые слова

Лесные дороги, спутниковые снимки, детекция дорог, определение центральных линий дорог, машинное обучение, семантическая сегментация, сбор данных, OpenStreetMap, географические данные.

1 Введение

Основной идеей проекта являлось получение карт лесных дорог по спутниковым снимкам. Современные карты, цифровые или бумажные, не обладают полной информацией о проселочных лесных дорогах, из-за чего могут возникать трудности с построением транспортных маршрутов. Однако, такие дороги довольно хорошо различимы на спутниковых снимках. Мы разработали модели машинного обучения, которые по спутниковым снимкам считывают сеть лесных дорог. Существуют работы по детекции дорог в сельской и лесной местностях [4, 9], но в отличие от нашей работы, их авторы не предоставляют открытого доступа к своим данным и моделям. В работе мы использовали данные из открытых источников, модели из популярных общедоступных библиотек глубинного обучения, библиотеки для работы с географическими данными и некоторые методы, изложенные в вышеупомянутых статьях по нашей теме.

Семантическая сегментация – присвоение каждому пикселию исходного изображения некоторого класса. Нам было необходимо разработать модель, которая будет сегментировать спутниковые снимки лесной местности, получать по ним маски изображений, на которых каждому пикселию будет присвоен класс 0 (фон) или класс 1 (дорога).

Многие работы [4, 9, 6, 10, 2, 5] в области извлечения карты дорог со спутниковых снимков применяют модели глубинного обучения, в частности сверточные нейронные сети, которые хорошо показали себя при работе с изображениями, их классификации, их семантической сегментации, детекции на них объектов. Для семантической сегментации снимков мы решили создать глубокую сверточную нейронную сеть на основе существующих state-of-the-art архитектур (таких, как FCN, U-Net, DeepLabv3). В ходе экспериментов мы подобрали архитектуру, выбрали модель с наибольшими значениями метрик на тестовых данных.

После получения бинарной карты дорог от нашей модели в растровом формате необходимо было преобразовать ее в векторный формат OpenStreetMap. Нами был написан класс ForestRoadsExtractor, который справляется с этой задачей.

Модели были написаны на языке python3 в pytorch, две были взяты из библиотеки torchvision и одна была написана самостоятельно. Наши модели добились следующего качества на тестовой выборке специально созданного в ходе курсовой работы датасета (см. Таблицу 4.1). Результаты, к сожалению, получились хуже, чем значения метрик для лучших моделей в статьях, посвященных детекции дорог в сельской/лесной местности, приведенных на графике (Рисунок 2.8).

Класс для обработки спутниковых снимков и перевода бинарной маски в формат osm

также был написан на python3.

Для взаимодействия пользователей с моделью было написано мобильное приложение на языке Java под android.

Цель проекта: создание модели машинного обучения, выявляющей лесные дороги по спутниковым снимкам; создание мобильного приложения для ее использования.

Задачи проекта:

- 1 Создание датасета лесных дорог.
- 2 Разработка нейронной сети.
- 3 Разработка алгоритма перевода полученной карты дорог в формат OpenStreetMap.
- 4 Разработка приложения на android для удобного взаимодействия пользователей с моделью.

Все задачи были решены. Задачами 1 - 3 занималась Ахматова Анна Увайсовна, студент группы БПМИ201. Задачей 4 занимался Степанов Артемий Романович, студент группы БПИ211.

2 Обзор литературы

Многие работы, посвященные распознаванию дорог и определению центральных линий дорог по спутниковым снимкам [4, 9, 6, 10, 2, 5], для семантической сегментации изображений используют модели глибного обучения, сверточные нейронные сети. Чтобы понять, как они устроены, были изучены несколько state-of-the-art архитектур.

2.1 Fully Convolutional Networks for Semantic Segmentation

В статье [3] авторы предлагают использовать для сегментации изображений нейронные сети, полностью состоящие из сверточных слоев. В своей работе они брали state-of-the-art сверточные нейронные сети для классификации (LeNet, AlexNet, VGG, GoogleNet), полностью связные слои в них переделывали в сверточные (конволюционные) и добавляли обратные сверточные слои (деконволюционные), которые инвертируют порядок действий в свертках и тем самым увеличивают размер изображений. Полученные сети работали уже неплохо, но

сегментации изображений все равно оставались грубыми. Чтобы исправить это, они решили модифицировать архитектуру, добавить связи, комбинирующие предсказания последнего слоя с картами признаков более ранних слоев (см. Рисунок 2.1). Так как выходы более ранних слоев видели меньше пикселей, их рецептивное поле меньше, то они точнее описывают соответствующие области исходного изображения. Добавление таких связей улучшает локализацию предсказаний и качество самой модели, что авторы продемонстрировали в ходе экспериментов.

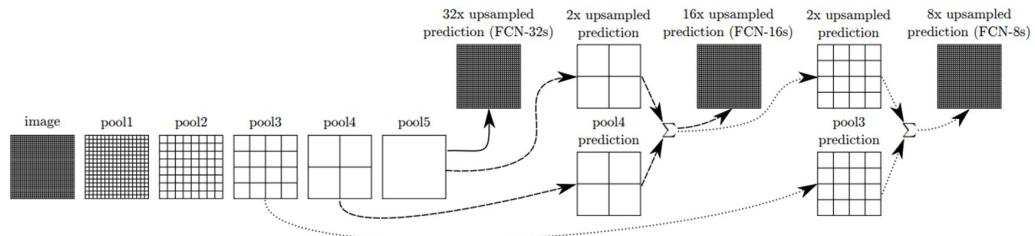


Рис. 2.1: Схема DAG из статьи [3].

2.2 U-Net: Convolutional Networks for Biomedical Image Segmentation

В статье [7] перед авторами стояла задача сконструировать сеть для семантической сегментации биомедицинских снимков, в частности снимков клеток. Они предложили следующую архитектуру (см. Рисунок 2.2). Сеть состоит из почти симметричных сужающегося и расширяющегося путей (отсюда U-форма и название U-Net). Основными ее особенностями являются, во-первых, использование up-convolution слоев (upsampling и convolution слой с ядром 2x2), которые увеличивают размерность, во-вторых, использование skip-connections, конкатенирование урезанных карт признаков из сужающегося пути с данными после up-convolution слоев расширяющегося пути, за счет чего получается лучшая локализация, более точная, аккуратная сегментация, в-третьих, постепенное увеличение и уменьшение количества каналов, которое дает возможность сети даже после сужающегося пути использовать весь полученный контекст и выявлять наиболее важные закономерности, зависимости в изображениях. Сеть показала хорошие результаты на биомедицинских датасетах. Архитектура стала state-of-the-art сетью и сейчас применяется для семантической сегментации изображений во многих областях.

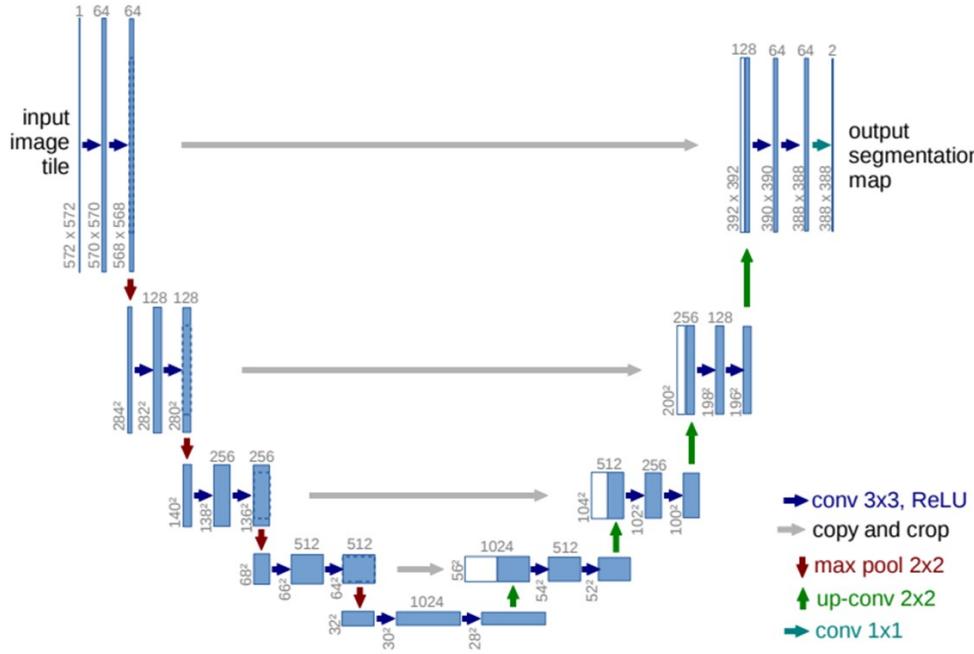


Рис. 2.2: Схема U-Net из статьи [7].

2.3 Rethinking Atrous Convolution for Semantic Image Segmentation

В статье [1] авторы выделяют две основные трудности в задаче семантической сегментации. Первая – сильное снижение размерности в результате последовательных pooling слоев и сверток с большими шагами, из-за чего теряется локализация, ухудшается качество предсказаний. Они предлагают бороться с этим с помощью atrous convolution слоев (или dilated convolution слоев) с увеличенными, разреженными пропусками, ядрами. Такие слои могут заменять последовательные pooling слои и свертки, так как они за счет пропусков увеличивают свое receptive field, захватывают больше информации/контекста без потребности в снижении размерности. Вторая сложность – то, что объекты на изображении могут быть разного размера, при определенном разрешении сеть может их уловить, при другом – уже не может. Чтобы бороться с этим, придумывают разные архитектуры, комбинирующие рассмотрение изображений в разных масштабах. Авторы статьи выделяют 4 основных (см. Рисунок 2.3): Image Pyramid (архитектура обрабатывает изображение в разных разрешениях, а потом комбинирует полученные карты признаков), Encoder-Decoder (использует карты признаков разных масштабов, полученные кодировщиком, и восстанавливает исходное разрешение декодировщиком), Deeper w. Atrous Convolution (дополнительные модули с atrous convolution накладываются поверх исходной сети для захвата информации, обобщающей изображение), Spatial Pyramid Pooling (SPP) (исходную карту признаков параллельно обрабатывают слоями с разными dilation, с разными receptive полями, затем агрегируют).

ют полученные результаты).

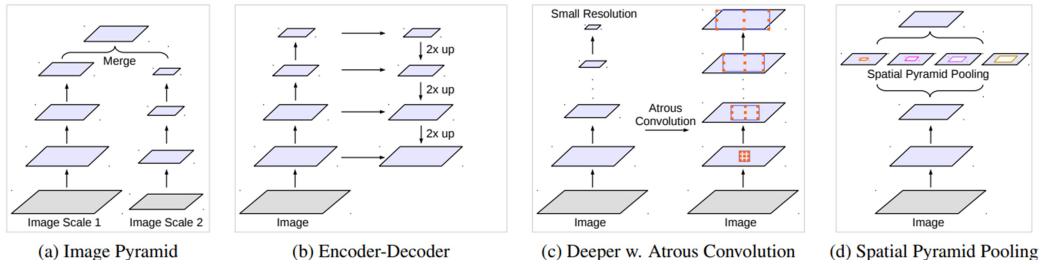


Рис. 2.3: Архитектуры, захватывающие контекст в разных масштабах, из статьи [1].

Авторы экспериментируют с atrous convolution и Atrous Spatial Pyramid Pooling. Модифицируют backbone ResNet, заменяя последние свертки в нем atrous convolution слоями и Atrous Spatial Pyramid Pooling.

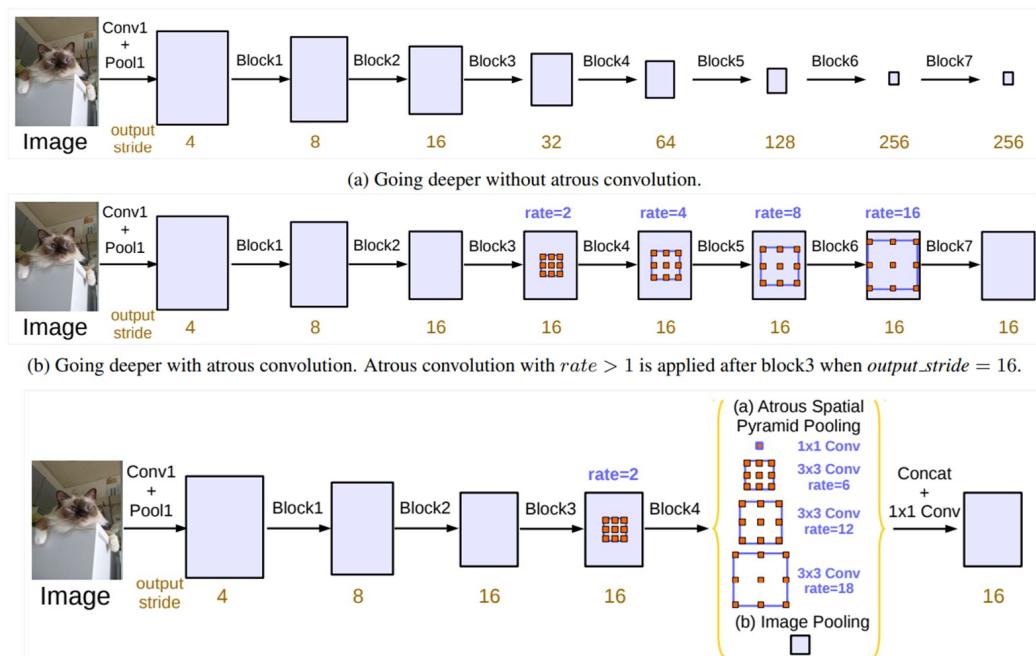


Рис. 2.4: Модификации backbone ResNet с помощью atrous convolution и ASPP в статье [1].

Авторы разработали сеть DeepLabv3, Encoder-Decoder типа, использующую atrous convolution, Atrous Spatial Pyramid Pooling, для семантической сегментации изображений. Результаты получились хорошими, на уровне с другими state-of-the-art сетями.

Нам удалось найти две работы, посвященные распознаванию дорог в сельской/лесной местности. Они сравнительно новые, первая вышла в январе 2023 года, вторая – в 2022 году.

2.4 Automatic Rural Road Centerline Detection and Extraction from Aerial Images for a Forest Fire Decision Support System

В работе [4] авторы поставили перед собой задачу разработать модель для детекции дорог в сельской местности по спутниковым снимкам и определения центральных линий этих дорог. Основной мотивацией было автоматизировать процесс построения карт дорог сельских местностей для использования пожарными, чтобы они могли быстро добираться до мест возгораний.

Модель, предложенная авторами, находит дороги по снимкам местности и достает из обрабатываемых изображений центральные линии этих дорог. Сначала она использует архитектуру DeepLabV3+ для сегментации изображения и получения бинарной маски, состоящей из фона (0) и дорог (1). Далее, чтобы получить информацию о центрах дорог, применяют алгоритм для исправления связности дорог и избавления от возможных артефактов (областей изображения, ложно классифицированных, как дорога). После этого методом скелетизации получают центральные линии дорог. Таким образом, подавая на вход спутниковый снимок, на выходе получают маску с черным фоном и тонкими, шириной в один пиксель, очертаниями центральных линий дорог.

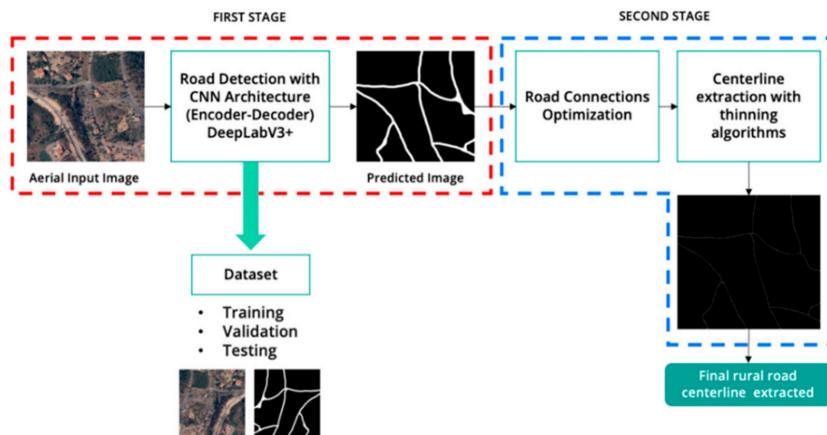


Рис. 2.5: Работа модели [4], сегментирующей спутниковые снимки и строящей центральные линии дорог.

Данные для обучения: 61 фото сельской местности в Португалии, высокого разрешения (пространственное разрешение 0.25 метров), в формате TIFF. Использовались RGB каналы изображений, каждое было размером 16000 пк x 10000 пк, покрывало местность 4 км x 2.5 км. Ground truth бинарные маски были сделаны вручную. Большие изображения и соответствующие им маски (16000x10000 пикселей) были разрезаны на небольшие части (1024x1024 пикселей), всего получилось 486 снимков. Так как данные собирались вручную,

выборка получилась небольшой, была произведена аугментация данных с помощью поворотов и отражений, всего получилось 3888 изображений.

В качестве архитектуры была выбрана DeepLabV3+ из-за использования в ней SPP (преимущество в том, что слой захватывает контекст разного разрешения/масштаба) и ее Encoder-Decoder архитектуры (которая хорошо показала себя в задачах семантической сегментации). Сеть обучалась, минимизируя dice loss, использовался оптимизатор Adam.

Полученные на выходе сети изображения затем трансформировались в бинарную карту дорог, ширина каждой из которых составляла 1 пк (см. Рисунок 2.6). Сначала дороги расширяли, восстанавливая связность. Затем, используя морфологические методы, уменьшали, делали шириной в 1 пиксель. В конце в цикле подсчитывались длины компонент связностей маски, короткие компоненты удалялись, как артефакты.



Рис. 2.6: Алгоритм из [4], восстанавливающий связность сети дорог и избавляющийся от артефактов, строящий центральные линии дорог.

Чтобы оценить модель, авторы рассматривали следующие метрики: completeness (COM), то же, что recall, correctness (COR), то же, что precision, quality (Q), то же, что IoU, и F1-score (F1).

$$COM = \frac{\text{Length of matched reference}}{\text{Length of reference}} \approx \frac{TP}{TP + FN} \in [0; 1] \quad (1)$$

$$COR = \frac{\text{Length of matched extraction}}{\text{Length of extraction}} \approx \frac{TP}{TP + FP} \in [0; 1] \quad (2)$$

$$Q = \frac{\text{Length of matched extraction}}{\text{Length of extracted data} + \text{Length of unmatched reference}} \approx \frac{TP}{TP + FN + FP} \in [0; 1] \quad (3)$$

$$F1 = \frac{2 \times COM \times COR}{COM + COR} \approx \frac{2 \times TP}{2 \times TP + FN + FP} \in [0; 1] \quad (4)$$

Здесь TP(true positive) - количество правильно распознанных пикселей дороги, TN (true negative) - количество правильно распознанных пикселей фона, FP (false positive) - количество пикселей фона, распознанных как дорога, FN (false negative) - количество пикселей дороги, распознанных как фон.

При этом, чтобы оценить качество определения центральных линий, эти метрики необходимо считать особым образом, так как линии очень тонкие, шириной всего один пиксель, и даже самые маленькие отклонения будут резко снижать значения. Для этого используют буфер фиксированного размера (см. Рисунок 2.7). Пиксели центральных линий, полученных моделью, которые попали в буфер, построенный вокруг референсных линий, считаются правильно выявленными (TP), те, что в буфер не попали, считаются неправильно выявленными (FP). Пиксели референсных центральных линий, которые не попали в буфер, построенный вокруг полученных моделью линий, считаются невыявленными (FN).

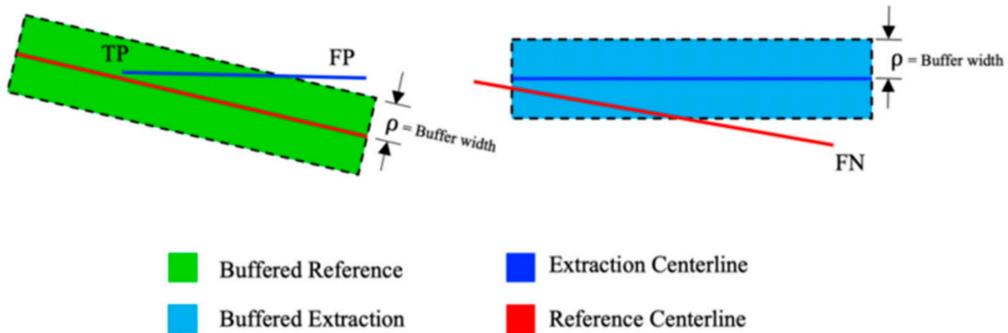


Рис. 2.7: Буферный метод подсчета метрик.

Авторы также провели эксперименты с сетями Unet и FPN, чтобы сравнить их с DeepLabV3+. Получили следующие результаты (см. Рисунок 2.8).

Авторы протестировали два морфологических алгоритма для сужения дорог: алгоритм Zhang–Suen [8] и алгоритм Guo-Hall [11].

В ходе экспериментов была построена модель с лучшими значениями метрик, она использовала DeepLabV3+, алгоритм Zhang–Suen, размер буфера – 8 пикселей. Значения

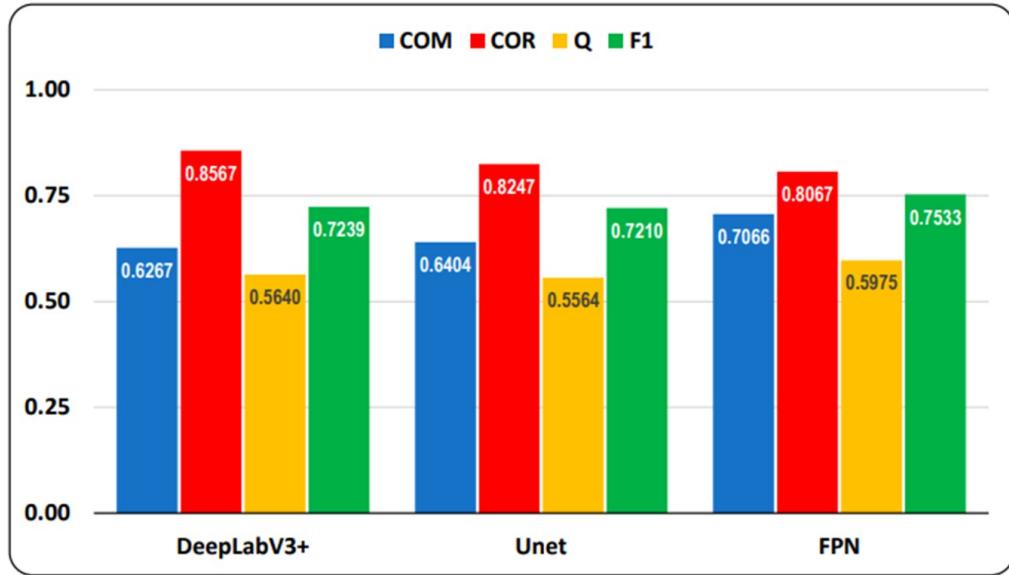


Рис. 2.8: Результаты [4], метрики на тестовом датасете для разных архитектур.

метрик: completeness = 0.8489, correctness = 0.8683, quality = 0.7588, F1 = 0.8585.

2.5 Detection of forest roads in Sentinel-2 images using U-Net

В статье [9] авторы поставили себе задачу разработать модель для детекции новых дорог в лесной местности. Они тоже сделали выбор в пользу глубокой нейронной сети, сегментирующей изображения.

Для обучения использовались мультиспектральные данные, полученные от спутника. Снимки были высокого разрешения (пространственное разрешение 10 метров), содержали 13 спектральных каналов с длинами волн от 492 нм до 2202 нм. Авторы использовали 10 спектральных каналов. Каждый снимок покрывал область 100 км x 100 км, представлял из себя изображение лесной местности в Норвегии. Всего в выборке присутствовало 38 снимков, 34 использовались для обучения, 2 – для валидации и 2 – для тестирования. Для получения ground truth изображений векторные данные о дорогах из открытых источников были переведены в растровые бинарные маски с разрешением 10 метров, покрывающие области 100 км x 100 км. Также были получены аналогичные маски для рек и озер, чтобы после получения масок дорог от сети исключать из них эти водные пространства, которые могут быть ложно распознаны, как дороги.

Авторы обучили сеть U-Net, минимизировали log loss, использовали оптимизатор Adam. У них получились следующие результаты на тестовом датасете (см. Рисунок 2.9), они исполь-

зовали метрики true positive rate = $\frac{TP}{(TP+FN)}$, false negative rate = $\frac{FN}{(TP+FN)}$, false positive rate = $\frac{FP}{(FP+TN)}$.

tile	date	roads			roads and tractor roads		
		true positive	false negative	false positive	true positive	false negative	false positive
T32VMN	2018-06-08	42 %	58 %	5 %	34 %	66 %	7 %
T32VMN	2018-07-03	52 %	48 %	6 %	41 %	59 %	7 %
T32VMN	2021-05-30	39 %	61 %	5 %	28 %	72 %	5 %
T32VMN	2021-07-22	45 %	55 %	7 %	36 %	64 %	7 %
T32VMN	2021-08-23	46 %	54 %	6 %	29 %	71 %	5 %
T32VNR	2018-05-09	37 %	63 %	5 %	25 %	75 %	6 %
T32VNR	2018-07-05	48 %	52 %	7 %	37 %	63 %	8 %
T32VNR	2019-04-24	30 %	70 %	4 %	20 %	80 %	5 %
T32VNR	2019-09-26	34 %	66 %	4 %	19 %	81 %	3 %
T32VNR	2020-06-19	52 %	48 %	8 %	41 %	59 %	13 %
T32VNR	2021-08-28	33 %	67 %	6 %	23 %	77 %	5 %
all test data		41 %	59 %	6 %	29 %	71 %	7 %

Рис. 2.9: Результаты на тестовом датасете [9].

Авторы также экспериментировали с использованием градиента цифровой модели рельефа. В работе не рассматривалась задача определения центральных линий дорог.

3 Создание датасета

В открытом доступе датасетов для семантической сегментации лесных дорог нами найдено не было. Нам пришлось создать собственный датасет, он получился небольшим.

Спутниковый снимок для датасета был взят из открытых источников, с сайта <https://scihub.copernicus.eu/dhus/#/home>. Он был получен спутником Sentinel-2, 25.08.2022. На нем изображена лесная озерная местность провинции Альберта Канады. Снимок покрывает участок площадью примерно 100 км x 100 км, координаты углов изображения: (55.94573440279175, -117.00032), (55.93320247103479, -115.24272), (54.947017819877004, -115.28601), (54.959098874363285, -117.000305). Он имеет разрешение 10 м, размер 10980 пк x 10980 пк. Для семантической сегментации было решено использовать изображения формата jp2, переведенные в более распространенный формат jpg, то есть мы работали с rgb-изображениями, имеющими 3 канала, которые соответствуют красному, зеленому и синему цветам.

Бинарная маска для снимка была сделана вручную, с помощью редактора GIMP. На белом фоне были размечены черные линии. Линии размечались на основе информации из карт OSM, Google Maps/Яндекс Карт. Дороги находились визуально, поэтому возможны погрешности и неточности. В ходе создания датасета мы убедились, что карты лесных дорог OSM действительно неполны (см. Рисунок 3.1).



Рис. 3.1: Часть спутникового снимка, карта OSM для него и карта после ручной разметки для него.

Была произведена аугментация данных. Снимок и маска были поделены на 3 части для обучающей, валидационной и тестовой выборок, примерно в отношении 70:15:15. В маске были изменены цвета: дороги стали обозначаться белым (класс 1), фон - черным (класс 0). По каждой части пускалось окно размером 500 пк x 500 пк (5 км x 5 км) с шагом 250 пк. Получившиеся изображения аугментировались поворотами (на 0, 90, 180, 270 градусов) и отражениями. Таким образом каждому кусочку размером 500x500 из исходного снимка соответствует 8 изображений в соответствующей выборке. Данные о соответствиях фотографий и масок сохранялись в csv-файлах. Размеры получившихся выборок (количество изображений): обучающей - 9744, валидационной - 1680, тестовой - 1680.

При таком наборе данных мы рассчитывали, что наша модель будет производить семантическую сегментацию изображений лесной местности в весенний/летний/осенний период (без снега), без помех в виде облаков.

Этой частью проекта занималась Ахматова Анна Увайсовна, студент группы БП-МИ201.

4 Подбор архитектуры нейронной сети

Были рассмотрены 3 архитектуры моделей для семантической сегментации: fcn_resnet50, unet и deeplabv3_resnet50. Имплементации fcn_resnet50 и deeplabv3_resnet50 были взяты из библиотеки torchvision, имплементация unet была написана самостоятельно.

4.1 Архитектура ResNet50

Backbone-моделью для архитектур fcn_resnet50 и deeplabv3_resnet50 является модель resnet50 для классификации изображений. Она является разновидностью моделей типа

ResNet, использует residual-соединения, то есть добавляет входные матрицы для сверточных блоков к их выходам, это помогает бороться с затуханием градиентов при обратном проходе во время оптимизации нейронной сети. Модель содержит 50 сверток, архитектура следующая:

- сначала применяется свертка с ядром 7×7 , количеством выходных каналов 64, шагом 2,
- затем применяется слой max pooling с ядром 3×3 , шагом 2,
- затем применяются 3 блока вида [свертка 1×1 , 64 -> свертка 3×3 , 64 -> свертка 1×1 , 256], размерность изображений не меняется, между блоками присутствуют residual-соединения,
- затем применяются 4 блока вида [свертка 1×1 , 128 -> свертка 3×3 , 128 -> свертка 1×1 , 512], размерность изображений уменьшается в 2 раза только в первой свертке первого блока за счет шага 2, между блоками присутствуют residual-соединения,
- затем применяются 6 блоков вида [свертка 1×1 , 256 -> свертка 3×3 , 256 -> свертка 1×1 , 1024], размерность изображений уменьшается в 2 раза только в первой свертке первого блока за счет шага 2, между блоками присутствуют residual-соединения,
- затем применяются 3 блока вида [свертка 1×1 , 512 -> свертка 3×3 , 512 -> свертка 1×1 , 2048], размерность изображений уменьшается в 2 раза только в первой свертке первого блока за счет шага 2, между блоками присутствуют residual-соединения,
- между блоками разного вида также присутствуют residual-соединения, там, где напрямую прибавить вход к выходу невозможно, вход проецируют на нужную размерность с помощью сверток 1×1 ,
- затем применяется слой average pooling,
- далее идет полносвязный слой, получают векторы размера, равного количеству классов,
- в конце применяют softmax.

В сетях для семантической сегментации модель модифицируют:

- 1 уменьшение размерности изображений с помощью шага 2 переносят с первой свертки в блоке на вторую, это повышает качество модели и проверено экспериментально,

2 после 3 первых больших сверточных блоков шаг 2 меняют на увеличение dilation в свертках в 2 раза, это позволяет сохранить локализацию.

4.2 Архитектура fcn_resnet50

Модель взята из библиотеки torchvision. Она использует в качестве backbone resnet50 с описанными выше модификациями, предобученный на ImageNet. В качестве head применяет свертку 3x3, без изменения размера (padding=1), с уменьшением количества каналов в 4 раза, батч-нормализацию, relu, dropout(0.1), свертку 1x1 с необходимым количеством выходных каналов.

Модель уменьшает размер изображений в 8 раз, чтобы его восстановить, к выходу модели применяется билинейная интерполяция.

4.3 Архитектура UNet

Имплементация UNet была написана самостоятельно, она почти в точности повторяет архитектуру из статьи [7]. Для упрощения мы в последовательных свертках между downsampling/upsampling не меняем размерность изображений, то есть применяем свертки 3x3 с padding=1, благодаря этому матрицы сужающегося пути при прибавлении к матрицам с расширяющегося пути можно не модифицировать.

4.4 Архитектура deeplabv3_resnet50

Модель взята из библиотеки torchvision. Она использует в качестве backbone resnet50 с описанными выше модификациями, предобученный на ImageNet. В качестве head применяет модуль ASPP, свертку 3x3, 256, батч-нормализацию, relu, свертку 1x1 с нужным количеством выходных каналов.

ASPP-модуль с параметрами dilation (12, 24, 36) представляет из себя следующее:

- используется блок [свертка 1x1, 256 -> батч-нормализация -> relu],
- используются блоки ASPPConv [свертка 3x3, 256, padding=dilation -> батч-нормализация -> relu] с dilation, равными 12, 24 и 36,
- используется блок ASPPPooling [adaptive average pooling (усреднение для каждого канала) -> свертка 1x1, 256 -> батч-нормализация -> relu -> билинейная интерполяция до входного размера],

- блоки применяются параллельно, они не меняют размер изображений,
- их выходы конкатенируются и проецируются на пространство с необходимым количеством каналов с помощью свертки 1×1 , затем идут батч-нормализация, `relu` и `dropout(0.5)`.

Модель уменьшает размер изображений в 8 раз, чтобы его восстановить, к выходу модели применяется билинейная интерполяция.

4.5 Результаты экспериментов

Датасет представляет из себя пары снимков и масок. Изображения хранятся в формате jpg, каждое можно представить в виде массива размера 3×500 пк $\times 500$ пк (3 канала $\times 5$ км $\times 5$ км, разрешение 1 пк – 10 м), заполненного целыми числами от 0 до 255. Перед подачей данных на вход модели снимки и маски переводятся в тензоры, увеличиваются до размера 512 пк $\times 512$ пк билинейной интерполяцией, снимки нормируются делением на 255, вычитанием 0.5 и делением на 0.5 (выравниваем значения на отрезок $[-1, 1]$), маски преобразуются в двумерные массивы, заполненные целыми числами, где в ячейке стоит 0, если пиксель – фон, и 1, если пиксель – дорога.

Подбирая модель, мы экспериментировали с аугментациями, оптимизаторами (`SGD / Adam`), их параметрами, функциями потерь (`CrossEntropyLoss / DiceLoss`), их параметрами. Для каждой архитектуры было проведено по 5 - 6 экспериментов. Как ни странно, наилучшие результаты для всех архитектур были получены при одних и тех же параметрах, описанных ниже.

Аугментации: для обучающего датасета после `Resize(512)`, перед нормированием снимков и обработкой масок производились аугментации: `RandomCrop(512, padding=64)` и `RandomRotation(45)`, также только для снимков применялось преобразование `ColorJitter(brightness=0.5, contrast=0.5, saturation=0.3, hue=0.3)`; для валидационного и тестового датасетов дополнительных аугментаций не было.

Оптимизатор: был использован оптимизатор `Adam` с `lr=0.00008, betas=(0.9, 0.999)`, `weight_decay=0.0001`.

Функция потерь: была использована функция потерь `DiceLoss (1 - F1)`.

Другие параметры: для ускорения обучения train выборка была обрезана до первых 8000 изображений для `fcn_resnet50` и `UNet`, до первых 6000 изображений для `deeplabv3_resnet50`; размер батча - 8; количество эпох - 14 для `fcn_resnet50`, 9 для `UNet`, 6 для `deeplabv3_resnet50`.

На Рисунке 4.1 представлены графики динамики loss, accuracy и метрик, обсуждавшихся ранее ([1](#), [2](#), [3](#), [4](#), без буфера), для train и validation выборок для наиболее удач-

ных экспериментов с этими тремя архитектурами. Значения для test выборки представлены в Таблице 4.1. Семантические сегментации, полученные моделями, для примеров из train, validation и test выборок показаны на Рисунке 4.2 для fcn_resnet50, на Рисунке 4.3 для UNet и на Рисунке 4.4 для deeplabv3_resnet50.

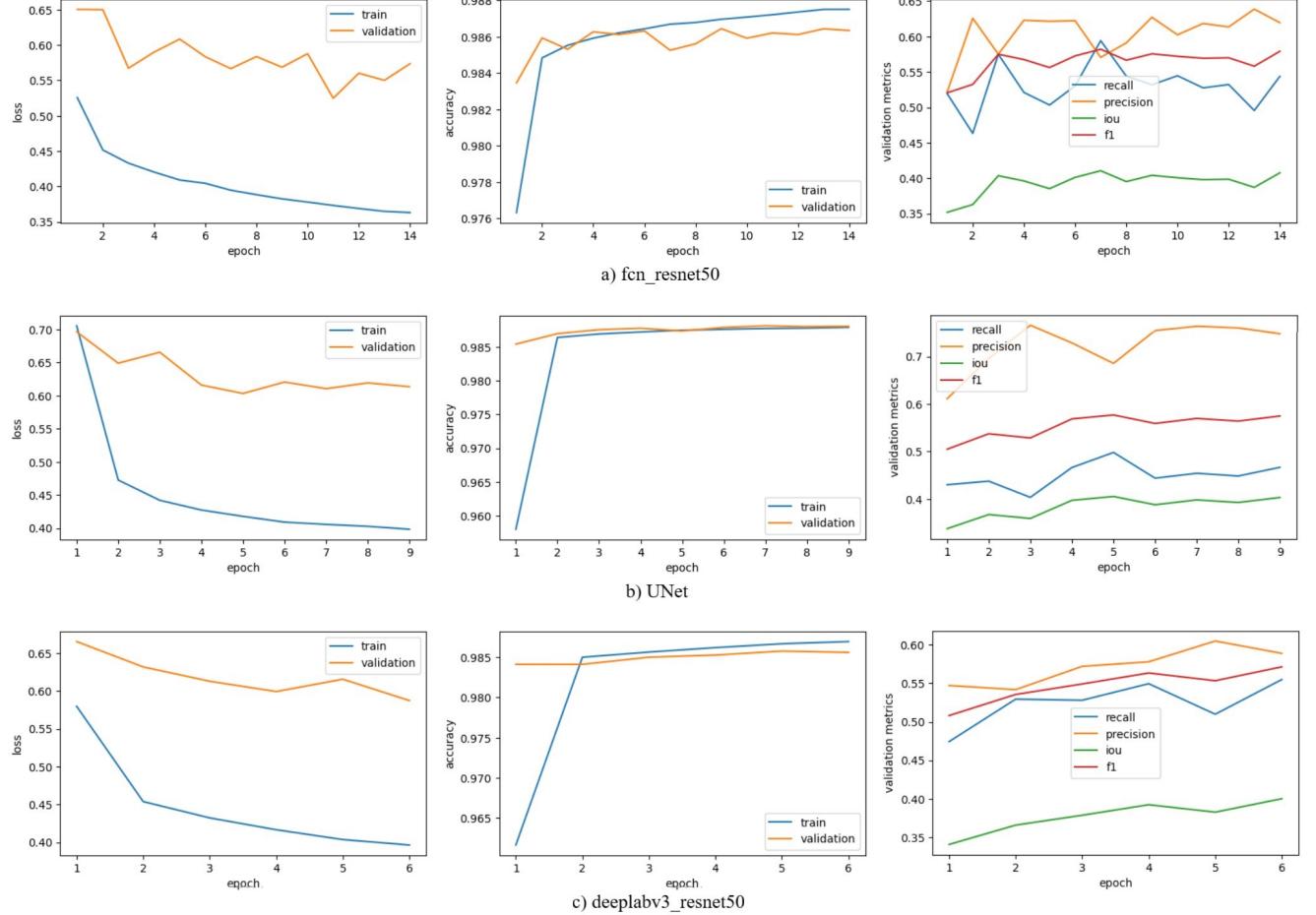


Рис. 4.1: Изменения loss, accuracy и метрик во время обучения для train и validation выборок для наилучших экспериментов с архитектурами.

Таблица 4.1: Значения accuracy и метрик для test выборки для наилучших экспериментов с архитектурами.

	Accuracy	Recall	Precision	IoU	F1
fcn_resnet50	0.98679	0.57628	0.60372	0.41812	0.58968
UNet	0.98881	0.52303	0.72088	0.43495	0.60622
deeplabv3_resnet50	0.98638	0.56402	0.59062	0.40550	0.57701

Наилучшее качество с точки зрения метрик получилось у модели UNet. Мы предлагаем использовать ее для семантической сегментации спутниковых снимков лесной местности и получения карт лесных дорог.

Обучение моделей производилось в средах kaggle и google colab с использованием GPU.

Одна эпоха каждого эксперимента занимала примерно 20 минут.

Этой частью проекта занималась Ахматова Анна Увайсовна, студент группы БП-МИ201.

5 ForestRoadsExtractor

Для взаимодействия с моделями, получения семантических сегментаций, их постобработки и преобразования в osm-файлы был написан класс ForestRoadsExtractor. Для его работы необходимы следующие библиотеки: pytorch, torchvision, numpy, opencv, osmnx. К нему прилагается код модели UNet, который был написан самостоятельно, в torchvision такой модели нет.

Методы класса ForestRoadsExtractor:

- `def __init__(self, model_name, model_path, device=torch.device('cuda:0' if torch.cuda.is_available() else 'cpu'))`
 - метод, инициализирующий класс, требует указания названия модели `model_name` ('fcn_resnet50', 'unet', 'deeplabv3_resnet50'), пути к файлу с моделью `model_path`, дополнительно можно указать `device`, на котором будут производится вычисления;
- `def get_raw_roads(self, image_name)`
 - метод, который строит маску для изображения, требует указания пути до файла `image_name` (изображение должно быть можно считать с помощью функции `torchvision.io.read_image`, изображение должно показывать лесную местность в весенний/летний/осенний период, без облаков, с разрешением 1 пк ~ 1 м), возвращает двухмерный массив с классами для пикселей (0 - фон и 1 - дорога); метод вызывает последовательно другие методы: `preprocessing`, `processing` и `postprocessing`; `preprocessing` делает `padding` изображения, чтобы его высота и ширина были больше 500 и делились на 250, проходит по нему с окном 500x500 с шагом 250, для кусочков делает `Resize(512)` и нормировку, сохраняет их в тензор `bigbatch`; `processing` пропускает `bigbatch` через модель, делает `argmax` для логитов и получает классы для пикселей, делает обратный `Resize(500)` и сохраняет кусочки в массив `bigbatch_masks`; `postprocessing` склеивает части вместе, если для пикселя модель хотя бы в одном из кусочков предсказала дорогу, то пикслю будет присвоен класс 1 в выходной маске, если во всех кусочках, покрывающих пиксель, модель предсказала для него фон, то пикслю будет присвоен класс 0, также метод делает `unpadding`;

- `def get_processed_roads(self, raw_roads)`
 - метод, обрабатывающий предсказанную маску, призванный улучшить качество выхода модели (аналогичен постобработке изображений из статьи [4]), требует на вход маску `raw_roads`, полученную методом `get_raw_roads`, возвращает обработанную маску и вспомогательные для построения векторной карты в формате osm словарь `inds_coords` (ставящий в соответствие номерам вершин в графе, которыми являются пиксели с классом 1, их координаты в маске) и график `graph` (списки смежности); метод вызывает последовательно другие методы: `dilation`, `skeletonization` и `get_rid_of_artifacts`; метод `dilation` расширяет дороги морфологическими методами (ядром 2x2) для возможного восстановления связности сети дорог; метод `skeletonization` сужает дороги так, чтобы они были шириной 1 пиксель, метод описан в статье [8], он итеративно убирает сначала правые нижние границы, затем левые верхние границы, пока находятся пиксели, подпадающие под условия удаления из изображения; метод `get_rid_of_artifacts` строит по получившейся маске график, он нумерует оставшиеся пиксели с классом 1, создает словарь `inds_coords`, считает эти пиксели вершинами и строит ребра между смежными пикселями дороги, обходит график в ширину, каждому пикселю дороги присваивает компоненту связности, затем компоненты связности мощности меньше 30 (длина дорог меньше 300 м) стирает с маски, как артефакты;
- `def get_osm_map(self, init_mask, inds_coords, graph, coords, osm_file_path)`
 - метод, строящий векторную карту в формате osm, требует на вход маску `init_mask`, словарь `inds_coords`, график `graph`, полученные методом `get_processed_roads`, а также координаты углов изображения `coords` в порядке: верхний левый, верхний правый, нижний правый, нижний левый, в формате (широта, долгота), и название для osm-файла `osm_file_path`; метод линейно вычисляет географические координаты всех пикселей изображения по координатам внутри массива и географическим координатам верхнего левого, верхнего правого и нижнего левого краев снимка; метод с помощью `geopandas` строит `GeoDataFrame nodes` для вершин и `GeoDataFrame edges` для ребер, с помощью функции `osmnx.utils_graph.graph_from_gdfs` строит по ним `networkx.MultiDiGraph G`, сохраняет его, как osm-файл, с помощью функции `osmnx.save_graph_xml`;
- `def build_osm_map(self, image_name, coords, osm_file_path)`
 - метод, который по спутниковому снимку и координатам его углов строит векторную карту дорог и сохраняет ее в osm-файл, требует указать путь к снимку `image_name`,

координаты углов изображения coords, название для osm-файла osm_file_path; метод последовательно вызывает другие методы: get_raw_roads, get_processed_roads и get_osm_map.

Примеры работы с классом представлены в репозитории. На Рисунке 5.1 изображены результаты обработки снимка классом.

Этой частью проекта занималась Ахматова Анна Увайсовна, студент группы БП-МИ201.

6 Анализ результатов

Чтобы оценить качество обученных моделей, мы решили выбрать 3 лесных участка и сравнить для них спутниковые снимки, карты из OSM и сегментации моделей. Результаты представлены на Рисунке 6.1.

Участки имеют координаты:

- первый - ((53.24962646775486, -117.000305),
(53.248851201797216, -116.88792487704917),
(53.18142768480264, -116.88792385245901),
(53.182202950760285, -117.00030397540984));
- второй - ((46.94616825397537, 22.314035280374384),
(46.944417443471046, 22.412501785544453),
(46.8769376801115, 22.410893096224296),
(46.87868849061583, 22.312426591054226));
- третий - ((57.74229815282701, 56.999664053339856),
(57.74138183324759, 57.125606764648786),
(57.67400835746346, 57.12560736950871),
(57.67492467704288, 56.99966465819978)).

Первый участок близок к местности, на основе которой построен наш датасет, поэтому семантические сегментации для него получились визуально довольно хорошими. Модели не смогли найти все дороги из OSM карт, но смогли найти новые, которые в OSM картах отсутствуют, но которые хорошо видно на снимке, а значит они большие, широкие.

Второй снимок изображает местность в восточной Европе. На нем даже визуально найти дороги достаточно сложно при таком разрешении. Модели плохо справились с этим

снимком, очень малую часть дорог из OSM карт им удалось найти. Скорее всего так получилось, потому что дороги в этой местности уже, чем в местности из датасета. Ситуацию потенциально можно исправить добавлением примеров таких сложных местностей в датасет, для этого требуется аккуратная ручная разметка, к сожалению, в ходе выполнения проекта этого сделать не удалось.

Третий снимок изображает местность на Урале. Модели справились с этим снимком лучше, чем с предыдущим. Они не смогли найти все дороги из OSM карт, но те, что хорошо видно на снимках, они смогли определить. Также, кажется, что модели fcn_resnet50 удалось найти несколько новых дорог в нижней части снимка.

Как видно по снимкам, работа моделей еще далека от идеала. Работу потенциально можно улучшить увеличением датасета более сложными примерами, использованием моделей с большим количеством параметров, с другими архитектурами. Как уже было сказано ранее, этого в ходе проекта сделать не удалось. Разметка даже небольшого датасета заняла более двух недель. Архитектуры, рассматриваемые нами в проекте, уже достаточно большие с учетом того, что мы планировали написать приложение под android для взаимодействия пользователей с моделью. Каждая модель имеет примерно 35000000 параметров, весит примерно 130 Мб. Стоит отметить, что модели поменьше, в частности deeplabv3_mobilenet_v3_large, дают результаты сильно хуже, чем модели побольше. Для deeplabv3_mobilenet_v3_large на валидационной выборке из нашего датасета IoU не смогло подняться выше 0.25.

Сравнивая результаты нашей работы с результатами статьи [4], можно сделать вывод, что значения метрик для тестового датасета, к сожалению, получились у нас ниже.

7 Приложение FOREST NAVIGATOR

Для удобного взаимодействия пользователей с моделью было разработано приложение под android на языке Java.

Приложение работает следующим образом:

- 1 Пользователь указывает местность путём перемещения камеры к необходимым координатам, выбирает масштаб и ориентацию участка карты, который далее передается нейронной сети.
- 2 Снимок передаётся нейронной сети. Модели на вход подается изображение размерами 500 пикселей x 500 пикселей с разрешением 1 пиксель – 1 м с 3 каналами rgб, переведенное в тензор, увеличенное до 512x512 и отнормированное. Модель на выходе выдает

логиты вероятностей классов пикселей. От них берется argmax, и так получается бинарная карта дорог.

3 Полученный результат представляется в виде растровой бинарной карты (0 – фон, 1 – дорога), которую приложение отображает дополнительным слоем поверх карты.

Приложение использует модель fcn_resnet50 с параметрами, при которых были достигнуты лучшие результаты в ходе экспериментов с этой архитектурой. Была выбрана эта модель, так как качество у нее ненамного хуже, чем у модели UNet, но у нее меньше параметров, она весит меньше и работает быстрее (fcn_resnet50 весит 126 Мб, обрабатывает 1 снимок на CPU за 2.84 с, UNet весит 131 Мб, обрабатывает 1 снимок на CPU за 6.45 с, deeplabv3_resnet50 весит 151 Мб, обрабатывает 1 снимок на CPU за 3.43 с).

Этой частью проекта занимался Степанов Артемий Романович, студент группы БПИ211.

8 Заключение

В этой курсовой работе перед нами стояла задача разработать модель машинного обучения для считывания сети лесных дорог со спутниковых снимков. Мы решали задачу семантической сегментации снимков лесной местности. В ходе работы мы создали новый датасет с лесными дорогами провинции Альберта Канады. Мы в ходе экспериментов опробовали несколько популярных state-of-the-art архитектур и подобрали глубокую нейронную сеть, достигшую наилучших результатов на тестовой выборке нашего датасета. Мы разработали алгоритм обработки снимков моделями, сохранения карт дорог в osm-формате и обернули его в класс ForestRoadsExtractor. Нами было написано приложение FOREST NAVIGATOR для взаимодействия пользователей с моделью. Все задачи курсовой работы выполнены. Все материалы выложены в открытый доступ в github-репозиториях и GoogleDrive.

Полученные модели получились неидеальными, они лучше работают в местности, близкой к местности созданного датасета, в других районах им достаточно сложно обнаружить даже те дороги, которые уже есть в OSM картах. Скорее всего, это происходит оттого, что в местности датасета дороги шире, чем в других. Этую проблему можно попытаться решить с помощью увеличения датасета. К сожалению, во время курсовой работы это сделать не получилось. Однако, дополнительно нужно учитывать тот факт, что спутниковые снимки в открытом доступе имеют достаточно низкое разрешение (наилучшее - 1 пиксель ~ 10 метров) и при таком разрешении даже человеку сложно разглядеть мелкие узкие проселочные дороги, модели сделать это тоже сложно. Несмотря на эти недостатки, модели все же могут

находить широкие лесные дороги, которых нет на OSM картах, и дополнять их, потому что эти карты действительно неполны.

Список литературы

- [1] Liang-Chieh Chen, George Papandreou, Florian Schroff и Hartwig Adam. “Rethinking Atrous Convolution for Semantic Image Segmentation”. B: *arXiv preprint, arXiv:1706.05587, version 3* (2017).
- [2] Ilke Demir, Forest Hughes, Aman Raj, Kaunil Dhruv, Suryanarayana Murthy Muddala, Sanyam Garg, Barrett Doo и Ramesh Raskar. “Generative Street Addresses from Satellite Imagery”. B: *International Journal of Geo-Information 7.3* (2018).
- [3] Jonathan Long, Evan Shelhamer и Trevor Darrell. “Fully Convolutional Networks for Semantic Segmentation”. B: *arXiv preprint, arXiv:1411.4038, version 2* (2015).
- [4] Miguel Lourenço, Diogo Estima, Henrique Oliveira, Luís Oliveira и André Mora. “Automatic Rural Road Centerline Detection and Extraction from Aerial Images for a Forest Fire Decision Support System”. B: *Remote Sensing 15.1* (2023).
- [5] Gellert M'attyus, Wenjie Luo и Raquel Urtasun. “DeepRoadMapper: Extracting Road Topology from Aerial Images”. B: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017.
- [6] Stefan Oehmcke, Christoffer Thrysøe, Andreas Borgstad, Marcos Antonio Vaz Salles, Martin Brandt и Fabian Gieseke. “Detecting Hardly Visible Roads in Low-Resolution Satellite Time Series Data”. B: *arXiv preprint, arXiv:1912.05026, version 1* (2019).
- [7] Olaf Ronneberger, Philipp Fischer и Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. B: *arXiv preprint, arXiv:1505.04597, version 1* (2015).
- [8] Zhang T.Y. и Suen C.Y. “A fast parallel algorithm for thinning digital patterns”. B: *Communication of the ACM 27* (1984), c. 236—239.
- [9] Øivind Due Trier, Arnt-Børre Salberg, Ragnvald Larsen и Ole T. Nyvoll. “Detection of forest roads in Sentinel-2 images using U-Net”. B: *Vol. 3 (2022): Proceedings of the Northern Lights Deep Learning Workshop 2022*. 2022.
- [10] Yao Wei, Kai Zhang и Shunping Ji. “Simultaneous Road Surface and Centerline Extraction From Large-Scale Remote Sensing Images Using CNN-Based Segmentation and Tracing”. B: *IEEE Transactions on Geoscience and Remote Sensing 58.12* (2020), c. 8919—8931.
- [11] Guo Z. и Hall R.W. “Parallel thinning with two-subiteration algorithms”. B: *Communications of the ACM 32* (1989), c. 359—373.

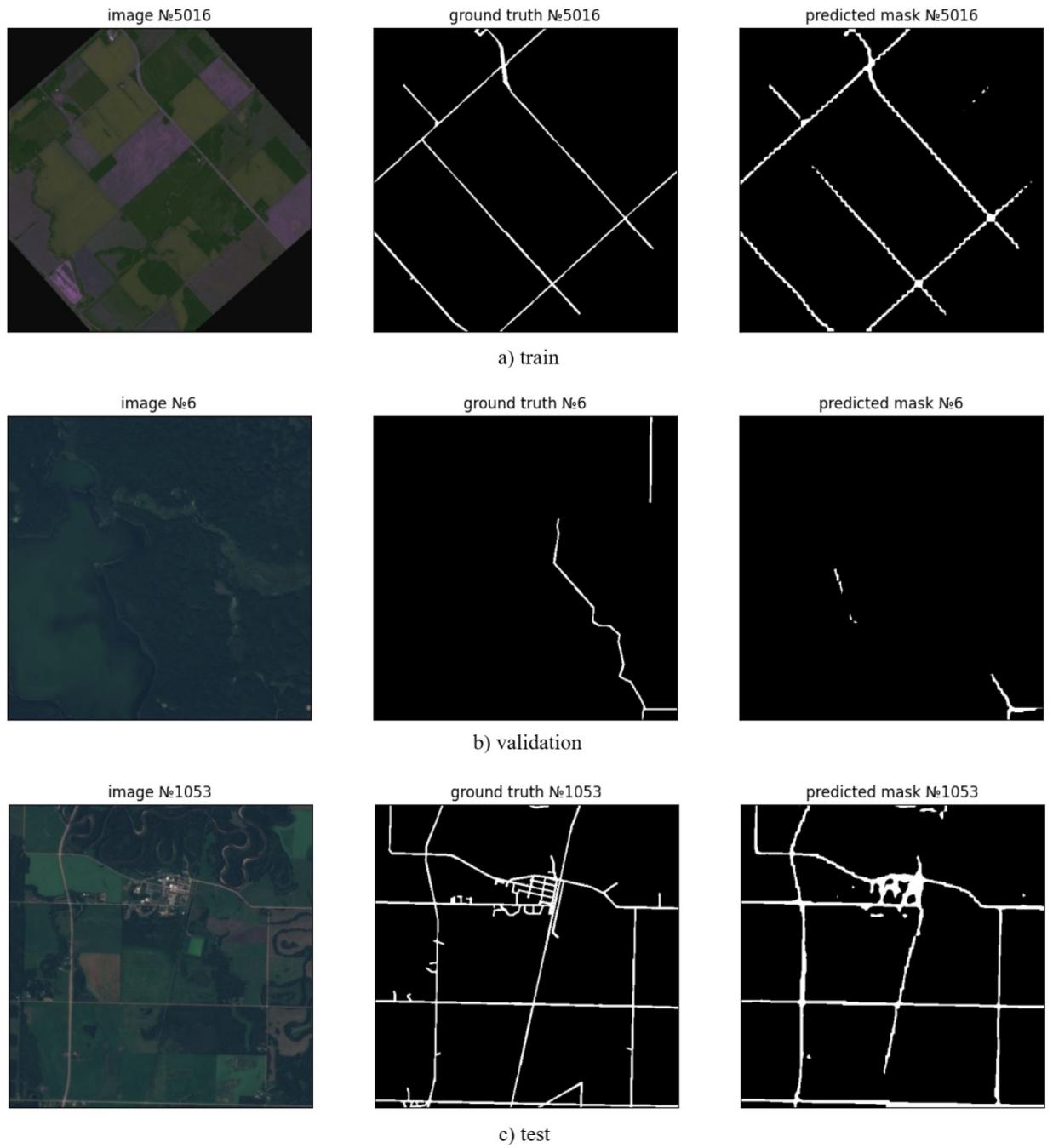


Рис. 4.2: Результаты модели fcn_resnet50.

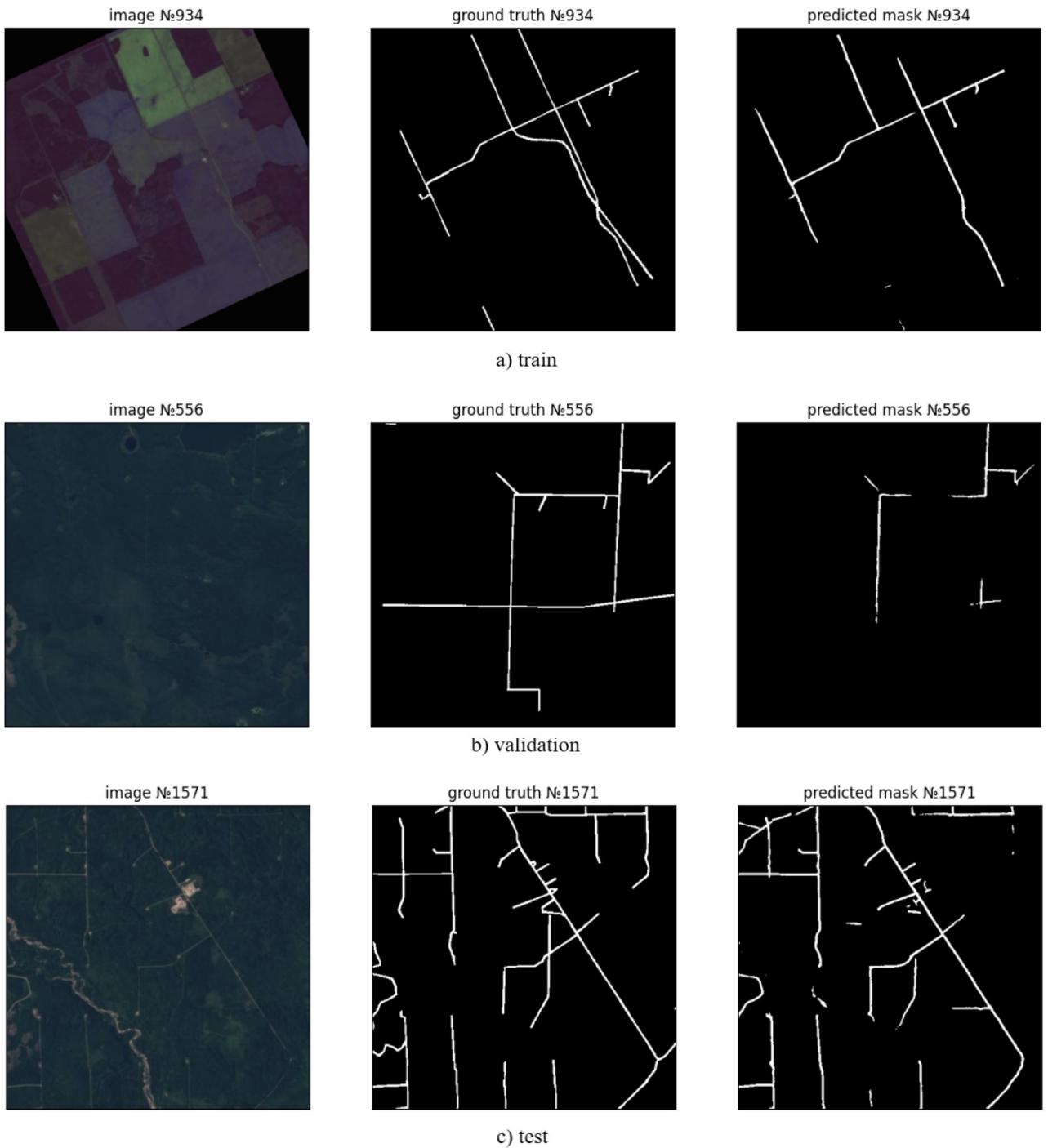


Рис. 4.3: Результаты модели UNet.

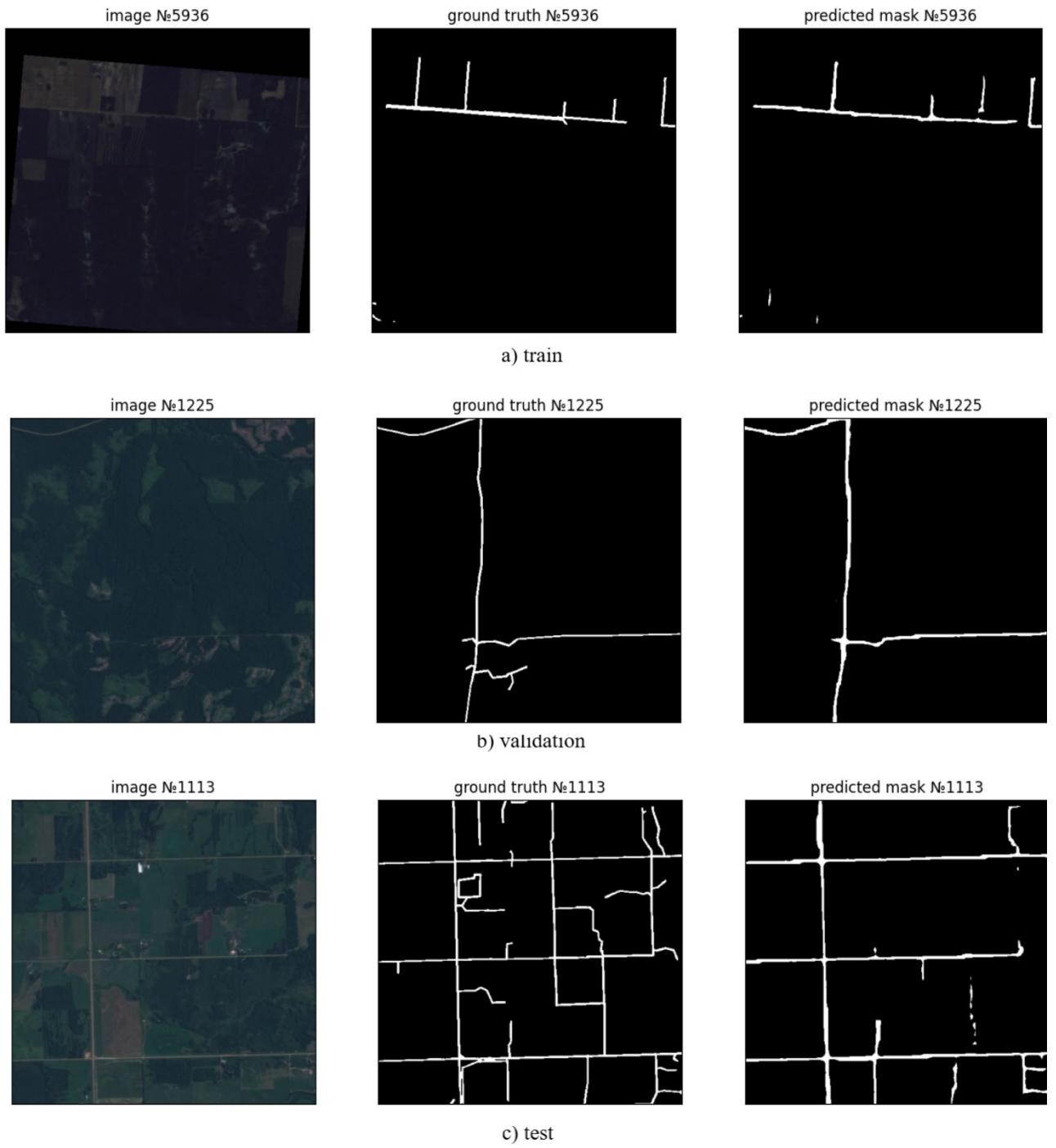


Рис. 4.4: Результаты модели deeplabv3_resnet50.

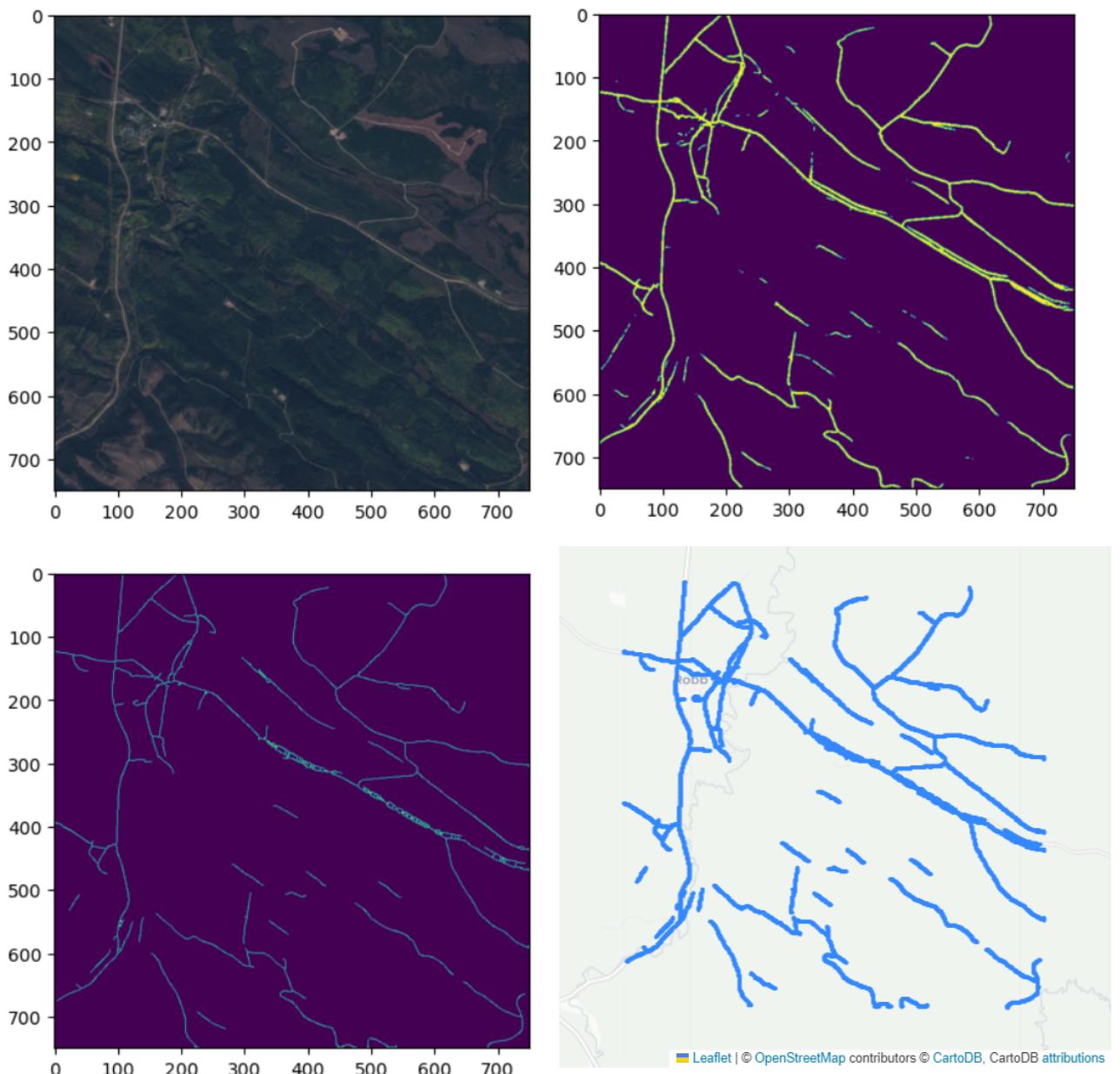


Рис. 5.1: Результаты обработки снимка классом ForestRoadsExtractor. Верхнее левое изображение - исходное, верхнее правое - результат вызова `get_raw_roads` (сырые предсказания), нижнее левое - результат вызова `get_processed_roads` (обработанные предсказания), нижнее правое - отрисованный граф (файл `osm` был преобразован в граф с помощью функции `osmnx.graph.graph_from_xml` и граф был отрисован в `folium` с помощью функции `osmnx.plot_graph_folium`).

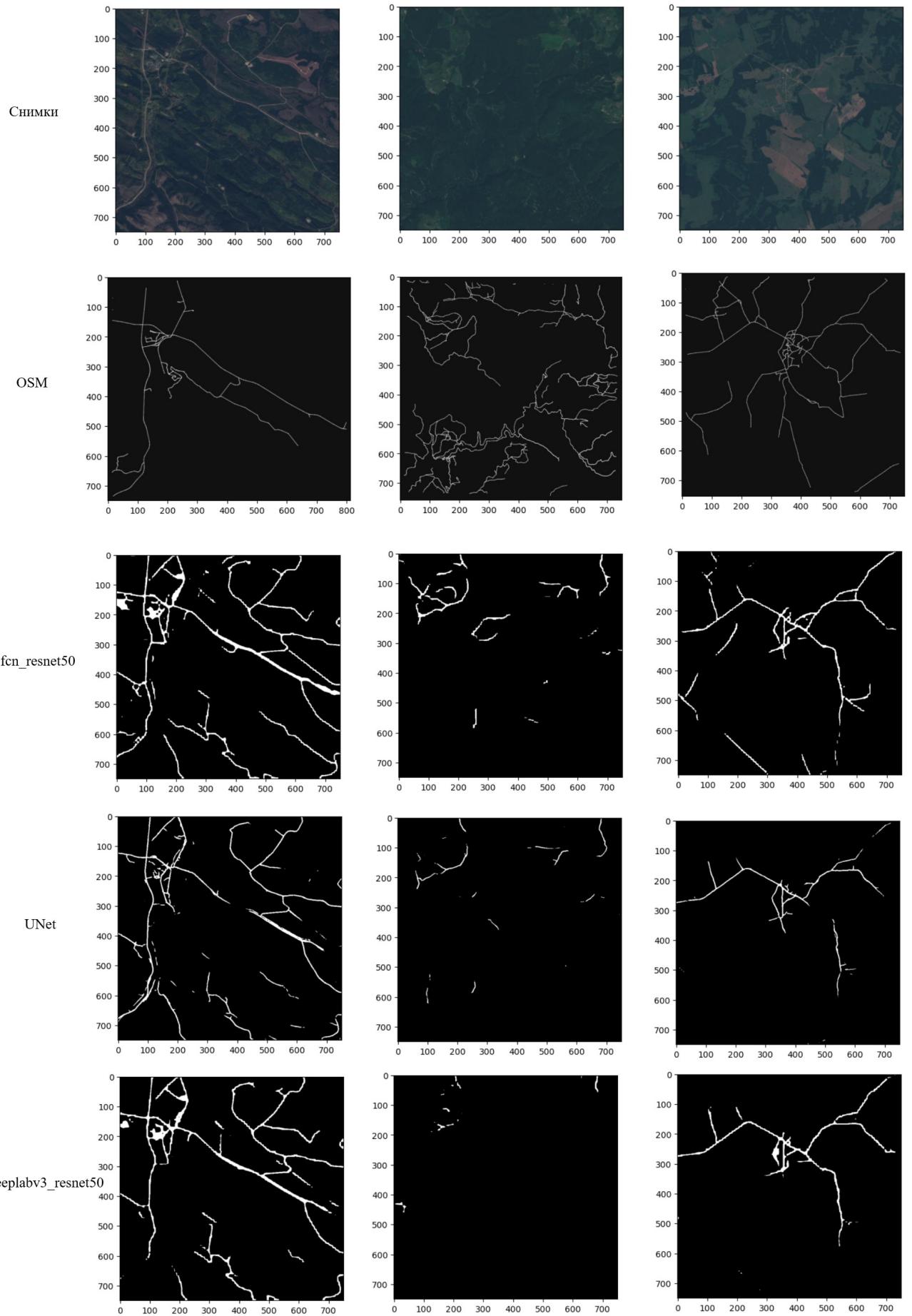


Рис. 6.1: Спутниковые снимки, карты OSM и сегментации, полученные моделями, для 3 спутниковых снимков лесной местности.