Научная статья
УДК 004.8
URL: https://trudymai.ru/published.php?ID=187461
EDN: https://www.elibrary.ru/HTPUOD

ПРИМЕНЕНИЕ НЕЙРОСЕТЕВЫХ МЕТОДОВ СЕМАНТИЧЕСКОЙ СЕГМЕНТАЦИИ В СИСТЕМАХ КОМПЬЮТЕРНОГО ЗРЕНИЯ РЕАЛЬНОГО ВРЕМЕНИ В УСЛОВИЯХ ОГРАНИЧЕННЫХ РЕСУРСОВ

Н.Г. Корыткин ✉

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет имени М.В. Ломоносова», г. Москва, Россия

✉ korytkinng@my.msu.ru

Цитирование: Корыткин Н.Г. Применение нейросетевых методов семантической сегментации в системах компьютерного зрения реального времени в условиях ограниченных ресурсов // Труды МАИ. 2026. № 146. URL: https://trudymai.ru/published.php?ID=187461

Аннотация. В работе исследованы две архитектуры нейронных сетей для семантической сегментации изображений в реальном времени на основе DeepLabv3+, в которых в качестве базовых моделей применены модифицированные MobileNetV3-Small и ResNet50. Для беспилотных транспортных средств (БТС), мобильных робототехнических комплексов (РТК) и аэрокосмических приложений, работающих в сложных и динамичных условиях, критически важно обеспечить высокую точность сегментации и скорость обработки видеопоследовательностей. Кодировщики модифицированы путем отбрасывания классификационных слоев с сохранением только сверточных слоев, отвечающих за извлечение признаков, что позволило интегрировать их в декодирующий модуль DeepLabv3+. В результате сформированы две архитектуры, различающиеся вычислительной сложностью и ориентированные на различные типы аппаратных платформ. Эксперименты проведены на двух тестовых стендах: настольном ПК с центральным процессором (ЦП) AMD Ryzen 5

3600 и дискретным графическим процессором (ГП) NVIDIA GeForce RTX 3050, а также на ноутбуке с мобильным процессором AMD Ryzen 7 5700U и интегрированным ГП. Обучение и валидация моделей выполнены на наборе данных Yamaha-CMU Off-Road (YCOR) с оценкой качества сегментации по метрикам mIoU, Pixel Accuracy и Mean Accuracy. Модель с кодировщиком MobileNetV3-Small продемонстрировала более высокое качество сегментации (mIoU = 55.56%) по сравнению с вариантом на базе ResNet50 (mIoU = 49.30%). В то же время архитектура с ResNet50 обеспечила более высокую производительность при использовании дискретного ГП. При наличии аппаратного ускорения обе модели достигли производительности не ниже 30 кадров в секунду при обработке видеопоследовательностей с разрешением 1920×1080 пикселей. Научная новизна работы заключается в детальном сравнении двух модификаций DeepLabv3+ с модифицированными кодировщиками MobileNetV3-Small и ResNet50 в условиях, приближенных к реальной эксплуатации мобильных робототехнических систем. Показано влияние типа аппаратной платформы на выбор архитектуры, обеспечивающей необходимый баланс между точностью сегментации и скоростью обработки видеопоследовательностей. На основе полученных результатов сформулированы практические рекомендации по применению разработанных моделей во встраиваемых и высокопроизводительных системах.

Ключевые слова: семантическая сегментация, DeepLabv3+, MobileNetV3, ResNet50, реальное время, робототехника, БТС.

APPLICATION OF NEURAL NETWORK-BASED SEMANTIC SEGMENTATION IN RESOURCE-CONSTRAINED REAL-TIME COMPUTER VISION SYSTEMS

N.G. Korytkin ✉

Federal State Budget Educational Institution of Higher Education M.V. Lomonosov Moscow State University, Moscow, Russia

✉ korytkinng@my.msu.ru

Citation: Korytkin N.G. Application of neural network-based semantic segmentation in resource-constrained real-time computer vision systems // Trudy MAI. 2026. No. 146. (In Russ.). URL: https://trudymai.ru/published.php?ID=187461

Abstract. This paper investigates two neural network architectures for real-time semantic image segmentation based on DeepLabv3+, employing modified MobileNetV3-Small and ResNet50 models as backbone encoders. For unmanned ground vehicles (UGVs), mobile robotic systems, and aerospace applications operating in complex and dynamic environments, achieving high segmentation accuracy and real-time processing performance is critically important. The encoders were modified by removing the classification layers while retaining the convolutional feature extraction layers, enabling their integration into the DeepLabv3+ decoder module. As a result, two architectures with different computational complexities were developed, each designed for specific hardware platforms. Experimental evaluation was conducted on two test platforms: a desktop system equipped with an AMD Ryzen 5 3600 CPU and an NVIDIA GeForce RTX 3050 discrete GPU, and a laptop featuring an AMD Ryzen 7 5700U mobile processor with integrated graphics. Training and validation were performed on the Yamaha-CMU Off-Road (YCOR) dataset using mIoU, Pixel Accuracy, and Mean Accuracy as evaluation metrics. The model with the MobileNetV3-Small encoder demonstrated superior segmentation accuracy (mIoU = 55.56%) compared to the ResNet50-based variant (mIoU = 49.30%). At the same time, the ResNet50 architecture achieved higher processing speed when executed on a discrete GPU. With hardware acceleration enabled, both models reached processing speeds of at least 30 frames per second for 1920×1080 video sequences. The scientific contribution of this work lies in a detailed comparative analysis of two modified DeepLabv3+ architectures under conditions approximating real-world deployment of mobile robotic systems. The influence of hardware platform type on the trade-off between segmentation accuracy and processing speed is demonstrated. Based on the obtained results, practical recommendations for selecting an appropriate architecture for embedded and high-performance systems are formulated.

Keywords: semantic segmentation, DeepLabv3+, MobileNetV3, ResNet50, real-time, robotics.

Введение

Современные мобильные роботы и беспилотные транспортные средства зависят от быстрого и точного распознавания объектов в реальном времени. Распространенность и доступность, а также удешевление камер высокого разрешения привели к возникновению большого количества «сырых» данных, для которых ручная разметка становится экономически нецелесообразной. Именно поэтому глубокие нейронные сети стали предпочтительным инструментом для автоматической семантической сегментации изображений.

Практическая применимость методов семантической сегментации подтверждается, например, в работе [1], где нейросетевая модель семантической сегментации земной поверхности используется для повышения точности позиционирования беспилотного летательного аппарата в реальном времени. Показано, что применение сверточной нейронной сети позволяет надежно выделять значимые элементы сцены, обеспечивая устойчивую навигационную информацию для аэрокосмических приложений. Более того, нейросетевые методы доказали свою эффективность при обработке различных типов изображений: распознавание распределенных объектов на радиолокационных изображениях [2], фильтрация шумов на радиолокационных изображениях [3]. Эти результаты демонстрируют, что нейронные сети способны эффективно выделять значимые признаки в сложных и шумных данных, что дополнительно мотивирует применение подобных подходов для семантической сегментации в мобильной робототехнике и беспилотных системах.

Семантическая сегментация – это задача компьютерного зрения, в которой каждое пиксельное значение изображения классифицируется по классу объекта (дорога, тропа, растительность и другие). Она объединяет два этапа: обнаружение сегментов и их последующую классификацию. Полученная карта сегментации служит основой для дальнейшего анализа сцены, планирования траектории робота и оценки проходимости дорожного покрытия.

Цель работы – исследование двух композитных архитектур на основе DeepLabv3+ для семантической сегментации изображений в реальном времени, применимых в мобильной робототехнике и беспилотных транспортных

средствах и аэрокосмических приложениях: модель 1 с модифицированным кодировщиком MobileNetV3-Small для энергоэффективных платформ и модель 2 с модифицированным кодировщиком ResNet50 для систем с дискретным ГП. В работе выполнена подготовка набора данных YCOR, обучение моделей и их оценка по ключевым метрикам точности: среднее пересечение по объединению (mean intersection-over-union, mIoU), пиксельная точность (Pixel Accuracy, PA), средняя точность (Mean Accuracy, MA), а также по показателю производительности – кадры в секунду (frames per second, FPS) на целевых платформах. Анализ полученных результатов позволяет найти и исследовать компромисс между точностью и скоростью работы моделей и выбрать оптимальную архитектуру, в зависимости от доступных вычислительных ресурсов.

Научная новизна работы состоит в проведении детального сравнения двух вариантов DeepLabv3+ с модифицированными кодировщиками MobileNetV3-Small и ResNet50 в условиях, приближенных к условиям реальной эксплуатации мобильных роботов, в которых важна скорость обработки и ограничены вычислительные ресурсы. Тестирование проводилось на трех типах аппаратных платформ: мобильном ЦП, настольном ЦП и дискретном ГП. Оценка моделей выполнялась как по стандартным метрикам сегментации, так и по таким показателям, как качество выделения дорожной области, точность определения ширины проезжей части и устойчивость к изменениям освещенности и погодных условий. Показано влияние типа аппаратной платформы на выбор архитектуры, обеспечивающей требуемый баланс между точностью сегментации и скоростью обработки кадров видеопоследовательности, на основе чего сформулированы практические рекомендации по применению моделей во встраиваемых и высокопроизводительных системах.

Постановка задачи

Семантическая сегментация – это задача компьютерного зрения, в которой происходит разбиения частей изображения на подгруппы пикселей, принадлежащих соответствующим объектам, с его классификацией.

Дано: RGB-изображение $X \in \mathbb{R}^{H \times W \times 3}$ (или одноканальное изображение).

Нужно получить карту сегментации $Y \in \mathbb{Z}^{H \times W}$, где каждый пиксель принадлежит одному из C классов.

С учетом ожидаемых скоростей движения робота от 5 до 40 км/ч требуется обработка видеопоследовательностей с частотой от 15 до 30 кадров в секунду, в зависимости от скорости движения и разрешением 1920×1080 пикселей.

Обзор существующих методов

Классические методы семантической сегментации изображений включают пороговую бинаризацию, наращивание областей и подходы, основанные на выделении границ. Пороговая бинаризация (thresholding) [4, 5] является одним из простейших методов, но малоэффективна для изображений со сложным распределением интенсивности из-за необходимости ручного выбора порогового значения. Метод наращивания областей (region growing) [4] хорошо работает с зашумленными изображениями, но его результаты существенно зависят от выбора начальных затравочных точек и критерия остановки. Методы, основанные на выделении границ (edge-based methods) [4, 6] эффективны при четких перепадах яркости, но чувствительны к шуму. На изображениях со сложной текстурой или плавными изменениями интенсивности границы оказываются фрагментированными, либо теряются, что затрудняет получение целостного контура объекта.

С развитием методов искусственного интеллекта появились более гибкие подходы к сегментации. Так, методы кластеризации (clustering-based methods) [4] отличаются простотой реализации и эффективностью, однако требуют заранее заданного числа кластеров. Наиболее перспективными на сегодняшний день являются нейронные сети, обеспечивающие высокую точность и способность обрабатывать большие объемы данных, но их применение связано с необходимостью обучения на подготовленных размеченных наборах данных.

На рисунке 1 приведен пример семантической сегментации изображения на наборе данных Cityscapes на классы: дорога, человек, автомобиль, здание и другие.

(a) (b)

Рисунок 1 – Пример сегментации. (a) – уличная сцена. (b) - семантически сегментированная уличная сцена

С развитием глубокого обучения предложен ряд специализированных архитектур для семантической сегментации. В 2015 году Джонатан Лонг, Эван Шелхаммер и Тревор Даррелл [7] представили Fully Convolutional Networks (FCN), полностью состоящую из сверточных слоев. Эта архитектура позволяет работать с изображениями произвольного размера, использует skip-connections (прямые соединения с ранними слоями сети, содержащими более детальную пространственную информацию) для улучшения точности. Однако обработка объектов разных масштабов затруднена (так как окно имеет фиксированный размер), что может приводить к грубой сегментации.

Стремление повысить детализацию сегментации привело к появлению DeconvNet [8], в которой кодировщик на основе VGG-16 дополнен симметричным декодировщиком с обратным пулингом и транспонированными свертками для восстановления пространственной информации. Данный подход позволяет лучше сохранять мелкие детали объектов по сравнению с FCN, но требует больших вычислительных затрат на обучение.

Компромиссное решение предложено в архитектуре SegNet [9], состоящей из кодировщика и декодировщика, за которым следует слой попиксельной классификации. SegNet работает медленнее FCN, но быстрее DeconvNet, поскольку отсутствуют полносвязные слои.

Для сегментации медицинских изображений разработана архитектура U-Net [10], использующая skip-connections и поэтапное восстановление пространственного разрешения из более грубых карт признаков (upsampling). Декодировщик симметричен кодировщику, что обеспечивает стабильность сегментации даже на небольших наборах данных.

Дальнейшее развитие методов семантической сегментации связано с серией моделей DeepLab (v1-v3+) [11, 12, 13, 14], которая развивается от пространственного пирамидального пулинга (SPP) до пространственной пирамиды сверток с пропусками (ASPP, Atrous Spatial Pyramid Pooling) и кодировщика-декодировщика в версии v3+, что обеспечивает постепенное улучшение точности сегментации. DeepLabv3+ является наиболее эффективной моделью среди представленных.

Предлагаемый метод

Выбор инструментальных средств

На основе проведенного обзора существующих методов сегментации изображений, включающего классические алгоритмы и нейросетевые подходы, в качестве основы выбрана архитектура DeepLabv3+ [14], являющаяся одной из наиболее эффективных в серии моделей DeepLab благодаря сочетанию модуля ASPP и структуры типа кодировщик-декодировщик. Оригинальная реализация DeepLabv3+ с модифицированным кодировщиком Xception содержит 47 миллионов обучаемых параметров [15], что влечет высокую вычислительную сложность при работе обученной модели и невозможность использования на устройствах с ограниченными ресурсами (например, на процессорах мобильных РТК или БЛА) в режиме реального времени. Более того, обучение такой архитектуры требует высокопроизводительных графических ускорителей, таких как NVIDIA Tesla V100, а также большое количество оперативной памяти, как, например, у авторов, имевших 27.4 ГБ [15].

Поэтому для возможности применения данной архитектуры в задаче семантической сегментации в реальном времени решено выполнить ее модификацию для снижения требований к вычислительным ресурсам и

оперативной памяти. В исходной архитектуре DeepLabv3+ кодировщиком выступает модифицированный Xception или ResNet101. В данной работе предложено заменить исходный кодировщик на более легкие архитектуры – MobileNetV3-Small и ResNet50. При этом архитектуры MobileNetV3-Small [16] (для модели 1) и ResNet50 [17] (для модели 2) дополнительно модифицированы путем отбрасывания классификационных слоев с сохранением только сверточных слоев, отвечающих за извлечение признаков, что позволяет интегрировать их в декодирующий модуль DeepLabv3+.

Следует отметить, что в последнее время одними из популярных и эффективных по затратам видеопамяти и вычислительной сложности являются архитектуры серии EfficientNet. Данная архитектура впервые представлена в 2020 году в статье [18], которая является попыткой достичь большей точности при снижении вычислительных затрат и объемов памяти. При этом авторы в данной статье не выполняют сравнение с существовавшей на тот момент MobileNetV3, которая появилась годом ранее. В 2021 году опубликована статья, в которой было представлено дальнейшее развитие архитектуры EfficientNet [19] под версией 2, но в работе так же не рассматривалась архитектура MobileNetV3. При этом в статье [20] представлено сравнение таких современных архитектур, как MobileNetV3-Small, ResNet18, EfficientNetV2, SqueezeNet, ShuffleNetV2. Показано, что MobileNetV3 на метрике F1-Score достигает 0.7188 при размере модели 7.46 МБ, а EfficientNetV2 достигает значения 0.7459 при размере 79.80 МБ. Таким образом, вес модели MobileNetV3 в 10.7 раз меньше, чем EfficientNetV2, уступая лишь на 3.63% на метрике F1-Score, но имея сопоставимый FLOPs. Кроме того, в статье [21] авторами также отмечено, что EfficientNetV2 достигает большой точности, в то время как MobileNetV3 предлагает лучший баланс между точностью и эффективностью при доминировании SqueezeNet в скорости обучения и компактности. Поэтому на основе изученной информации в данных статьях оптимальным считаем выбор MobileNetV3-Small как архитектуры, предлагающей лучший баланс для применения на устройствах с ограниченными вычислительными ресурсами.

Помимо этого, считаем целесообразным замену исходного кодировщика, модифицированного Xception (или ResNet101) в архитектуре DeepLabv3+, на более легкий вариант, ResNet50, для использования в качестве некоторого контрольного варианта на случай получения аномальных результатов. Поскольку в проанализированных работах архитектура MobileNetV3 часто остается без внимания, можем предположить, что MobileNetV3 покажет себя не лучшим образом на валидации.

В результате выполненных модификаций получены две композитные архитектуры сегментации: модель 1 (DeepLabv3+ с кодировщиком на базе модифицированной сети MobileNetV3-Small), модель 2 (DeepLabv3+ с кодировщиком на базе модифицированной сети ResNet50). Основные характеристики моделей 1 и 2 приведены в таблице 1.

Для разработки использовался язык программирования Python и открытый фреймворк для глубокого обучения TensorFlow.

Таблица 1

Сравнение моделей 1 и 2 по ключевым параметрам

Параметр	Модель 1	Модель 2
Структура	кодировщик-декодировщик	кодировщик-декодировщик
Тип кодировщика	MobileNetV3-Small	ResNet50
Тип декодировщика	DeepLabv3+ head	DeepLabv3+ head
Параметры кодировщика	0.94 млн	8.32 млн
Параметры декодировщика	5.91 млн	3.53 млн
Общее количество параметров модели	6.85 млн	11.85 млн
Целевая аппаратная платформа	Бортовые вычислительные системы мобильных РТК	Высокопроизводительные вычислительные системы с дискретными графическими ускорителями

В качестве основы для обучения использовался набор данных YCOR [22], содержащий 1076 изображений с разметкой по 8 классам: небо, неровная тропа, гладкая тропа, проходимая трава, высокая растительность, непроходимая низкая растительность, препятствие. Изображения разрешения 1024×544, собраны при скорости движения 5 км/ч по пересеченной местности в трех временах года (рисунок 2).

Рисунок 2 – Кадры из набора данных YCOR

Авторами набора данных случайным образом проведено разбиение изображений на два набора:

1. Обучающий набор состоит из 931 изображения.
2. Валидационный набор состоит из 145 изображений.

Используя [23], преобразуем набор данных YCOR к стандарту ADE20K. Изменится структура каталогов, появятся новые каталоги «annotations» и «images» в формате ADE20K.

Палитра 9 классов набора данных YCOR (8 оригинальных классов + фоновый класс), приведенного к стандарту ADE20K, изображена на рисунке 3.

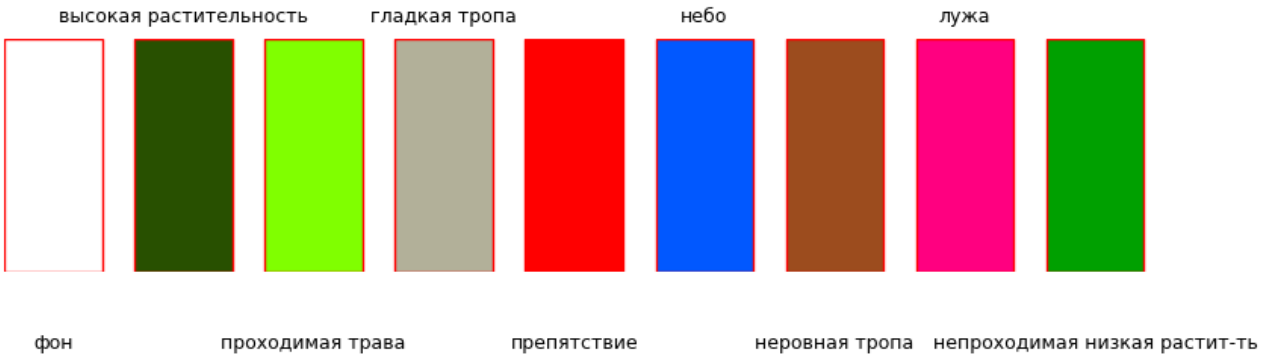


Рисунок 3 – Классы в наборе данных YCOR

На рисунке 4 представлена архитектура предложенной модели 1 – DeepLabv3+ с модифицированным кодировщиком MobileNetV3-Small. Модель состоит из трех основных компонентов: кодировщика (backbone), модуля ASPP и декодировщика. Кодировщик MobileNetV3-Small построен на основе

инвертированных bottleneck-блоков (Mobile Inverted Bottleneck Convolution, MBConv), которые могут включать механизм канального внимания Squeeze-and-Excitation (SE). В архитектуре используются четыре типа MBConv-блоков (рисунок 5), различающихся наличием SE-механизма, residual-соединений и операцией понижения пространственного разрешения (downsampling). На рисунке 6 представлены: (a) структура модуля ASPP, (b) архитектура блока декодировщика.

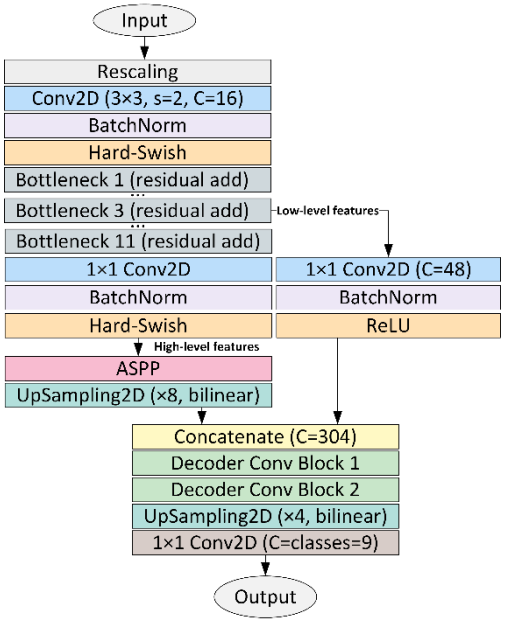


Рисунок 4 –Архитектура модели 1

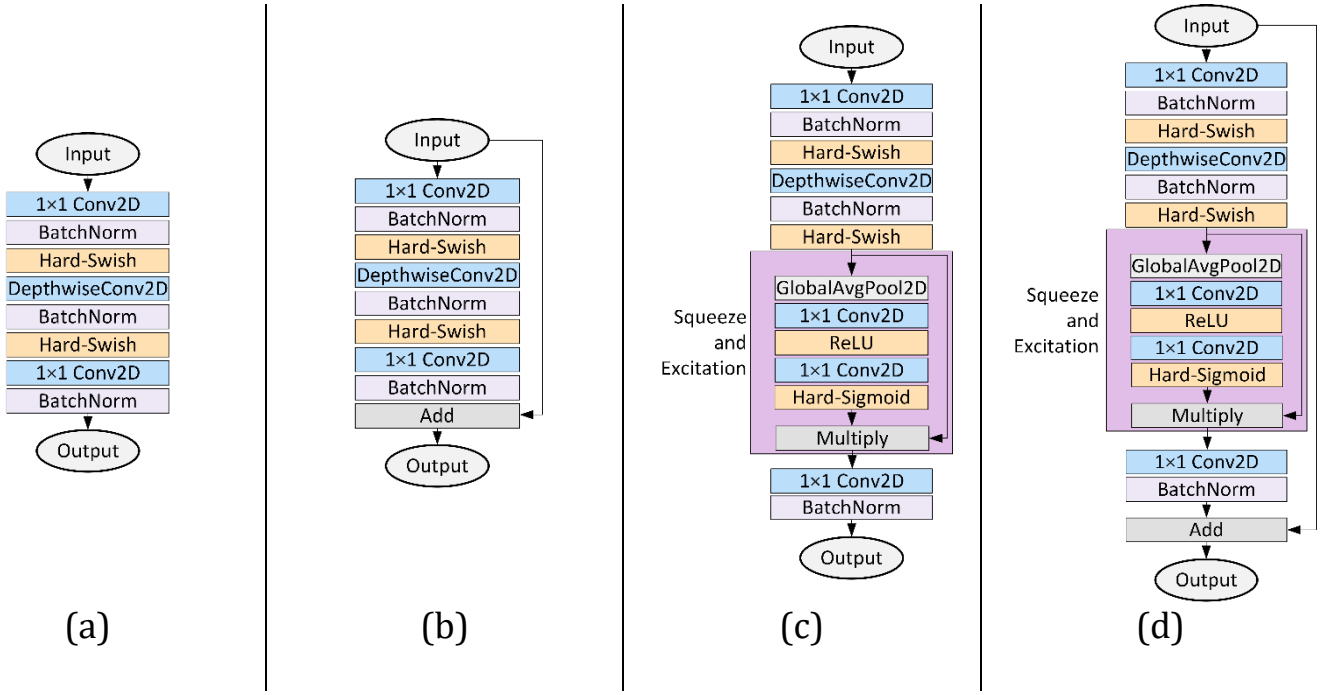


Рисунок 5 – Модель 1, типы MBConv-блоков в кодировщике: (a) – без SE и skip-connection, с downsampling, (b) – без SE, со skip-connection, (c) с SE, без skip-connection, с downsampling, (d) – с SE и skip-connection

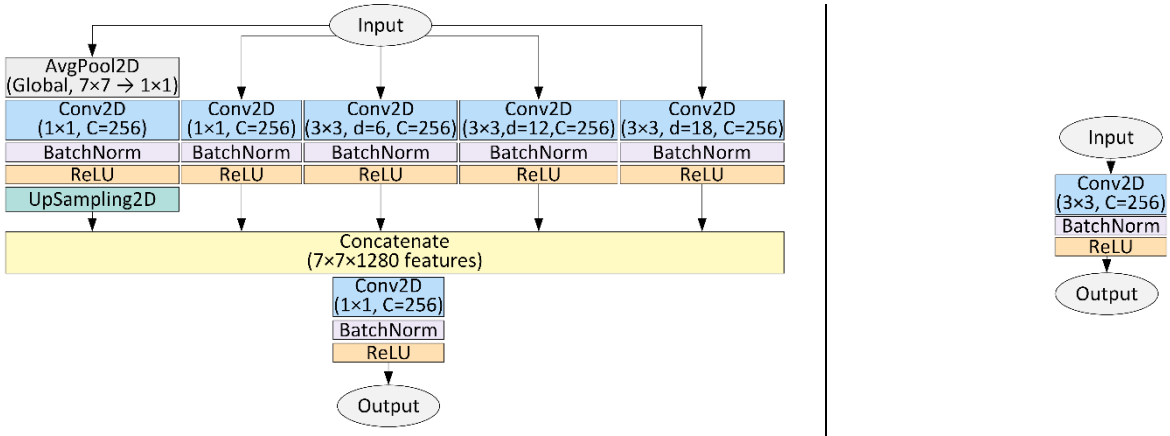


Рисунок 6 – Модель 1, (a) – структура модуля ASPP, (b) – структура Decoder Conv Block

На рисунке 7 – (a) представлена архитектура предложенной модели 2 – DeepLabv3+ с кодировщиком ResNet50. Как и модель 1, данная архитектура состоит из трех основных компонентов: кодировщика (backbone), модуля ASPP и декодировщика. Кодировщик ResNet50 построен на основе bottleneck-блоков с residual-соединениями (shortcut connections). В архитектуре используются два типа residual-блоков (рисунок 7 – (b), (c)). Блок с проекционным shortcut (projection shortcut) применяется при изменении размерности, для этого shortcut-путь включает свертку 1×1 для согласования размерностей входного и выходного тензоров. Блок с identity shortcut используется при сохранении размерности, где shortcut-путь представляет собой прямое тождественное соединение без дополнительных преобразований. Особенностью ResNet50 в модели 2 является иерархическая структура, где bottleneck-блоки объединены в три последовательных этапа (stages): на первом используются 3 блока, на втором – 4 блока, на третьем – 6 блоков. Каждый этап работает с признаками определенного пространственного разрешения. На рисунке 8 представлены: (a) – структура модуля ASPP, (b) – архитектура блока декодировщика.

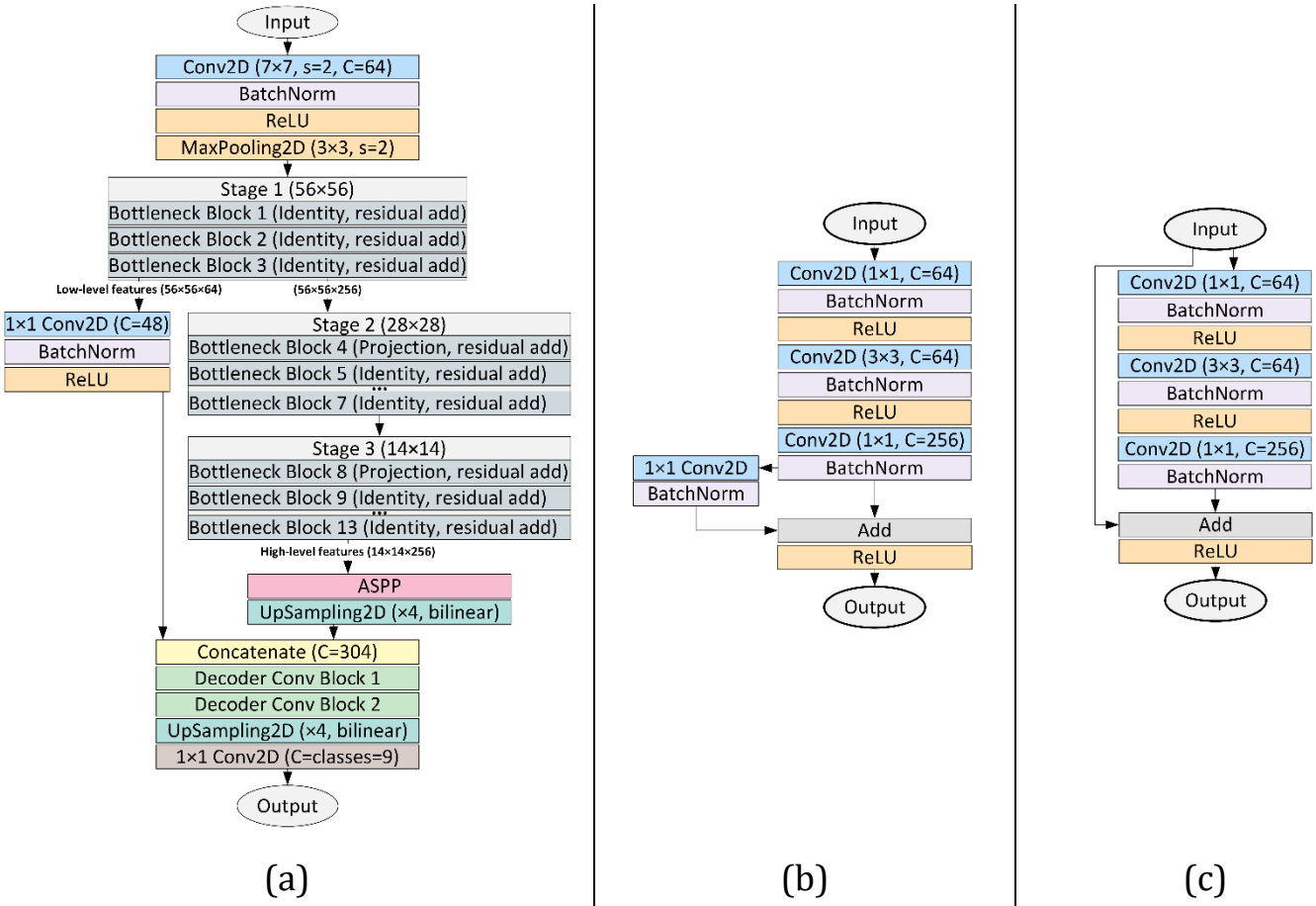


Рисунок 7 – Архитектура модели 2: (a) – общая схема; (b)-(c) типы bottleneck-блоков: (b) – проекционный, (c) – identity

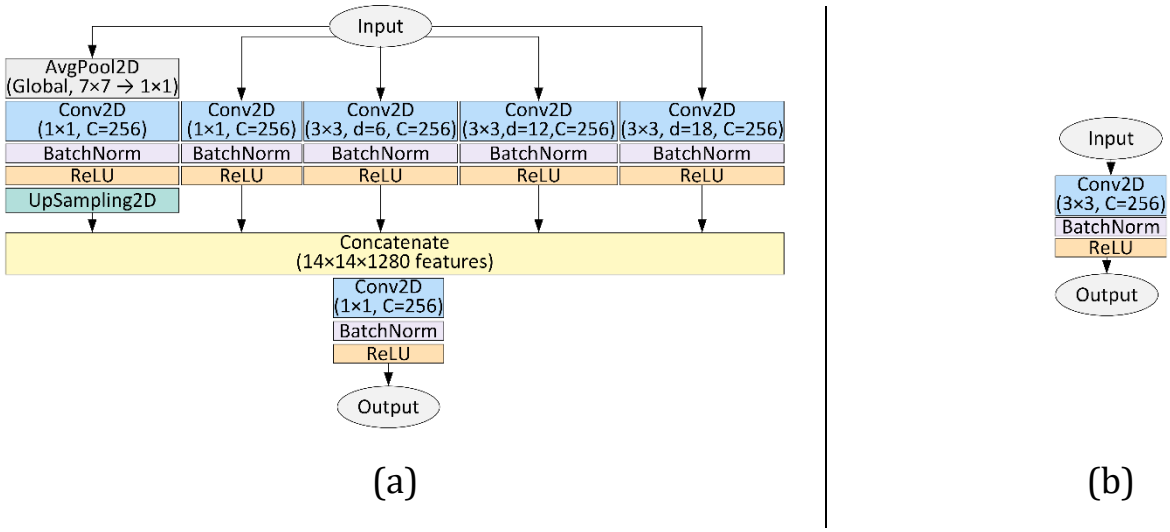


Рисунок 8 – Модель 2, (a) – структура модуля ASPP, (b) – структура Decoder Conv Block

В качестве основы использовались веса моделей MobileNetV3-Small и ResNet50, предварительно обученных на наборе данных ImageNet. Далее выполнено дообучение двух разработанных архитектур нейронных сетей на подготовленном наборе данных YCOR.

При обучении применялось адаптивное снижение скорости обучения при обнаружении плато функции потерь на валидационном наборе данных. Размер пакетной выборки (batch size) составлял 16, максимальное число эпох обучения – 500.

Поскольку решалась задача семантической сегментации, в которой каждому пикселю изображения соответствует ровно один класс, в качестве функции потерь использована Sparse Categorical Cross Entropy. В качестве оптимизатора применялся NAdam, представляющий собой комбинацию оптимизатора Adam и метода ускоренного градиентного спуска Нестерова.

Обучение выполнялось на графическом процессоре NVIDIA GeForce RTX 3050 (8 ГБ видеопамяти).

Для тестирования использовались видеопоследовательности с разрешением 1920×1080 пикселей и частотой 30 кадров в секунду, полученные из открытых источников и содержащие сцены движения по пересеченной местности при скоростях 5, 10, 16, 30 км/ч.

Эксперименты и результаты

Конфигурация тестовых стендов

Для оценки производительности использовались два тестовых стенда, их характеристики представлены в таблице 2.

Таблица 2

Конфигурация тестовых стендов
для тестирования производительности разработанных архитектур нейронных сетей

Характеристика	Стенд 1 (настольный ПК)	Стенд 2 (ноутбук)
Центральный процессор	AMD Ryzen 5 3600	AMD Ryzen 7 5700U
Графический процессор	NVIDIA GeForce RTX 3050	интегрированный AMD Radeon Graphics
Оперативная память	80 ГБ ОЗУ	32 ГБ ОЗУ
Операционная система	Windows 10 Pro 64-bit	Windows 10 Pro 64-bit

Дискретный ГП NVIDIA GeForce RTX 3050 выбран для оценки производительности моделей, так как по вычислительной мощности (TFLOPS) RTX 3050 сопоставима с мобильными платформами NVIDIA Jetson T4000 и T5000, предназначенными для применения в беспилотных летательных аппаратах

(БЛА) и мобильной робототехнике. Это позволяет проводить тестирование моделей в условиях, близких к реальным встраиваемым системам с ГП.

Мобильный процессор AMD Ryzen 7 5700U выбран для оценки работы моделей на устройствах без дискретного ГП. Он сочетает энергоэффективность и высокую производительность на ядро, что позволяет моделям работать на портативных и встраиваемых системах, включая БЛА и мобильных роботов, где важны энергопотребление, компактность и ограниченные вычислительные ресурсы.

Анализ процесса обучения

Для оценки сходимости моделей и выявления возможного переобучения фиксировались значения функции потерь (loss) и точности (accuracy) на обучающей и валидационной выборках. Для модели 1 (DeepLabv3+ с MobileNetV3-Small) и модели 2 (DeepLabv3+ с ResNet50) на рисунках 9 и 11 представлены графики функции потерь, на рисунках 10 и 12 представлены графики точности.

1. Функция потерь (рисунки 9 и 11). На обучающей выборке с увеличением числа эпох значение функции потерь уменьшается. Это означает, что модель успешно обучается. На валидационной выборке с ростом числа эпох значение функции потерь сначала уменьшается, затем снова начинает расти. После этого снова значение функции потерь уменьшается, а затем снова немного растет.

2. Точность (рисунки 10 и 12). На обучающей выборке точность сначала падает, затем начинает расти с увеличением числа эпох. Это свидетельствует о том, что модель с каждой эпохой лучше сегментирует данные в процессе обучения. На валидационной выборке точность растет с незначительными колебаниями в сторону увеличения и уменьшения, но с увеличением числа эпох значение точности растет.

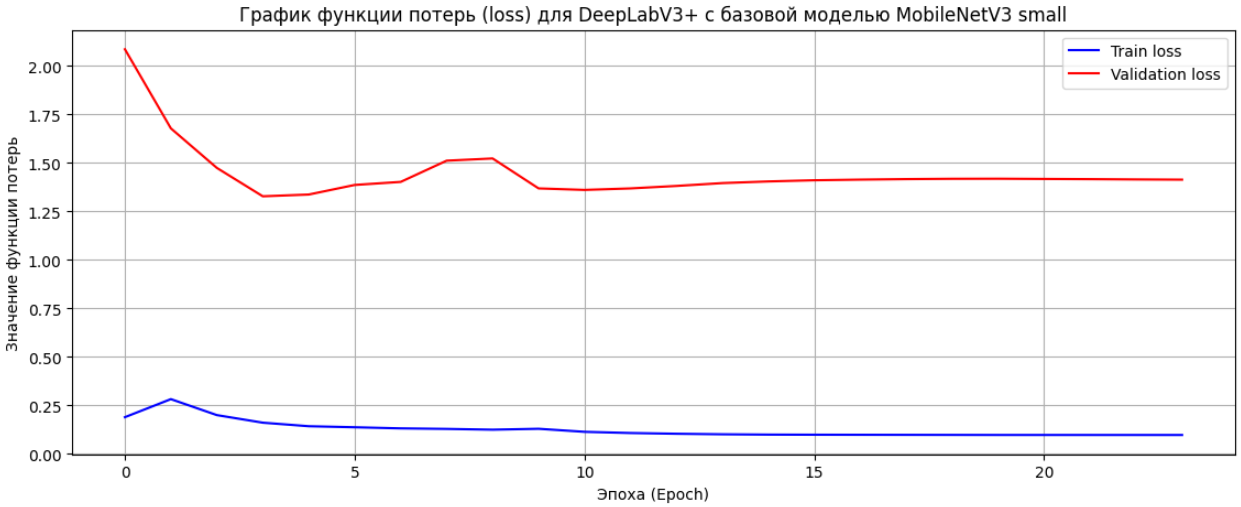


Рисунок 9 – График функции потерь для модели 1

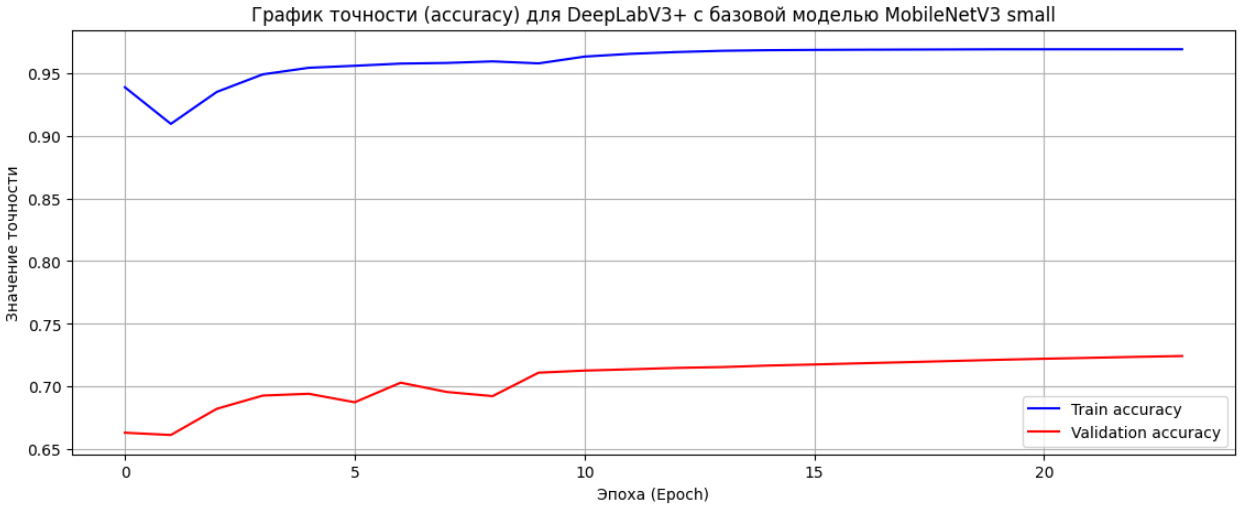


Рисунок 10 – График точности для модели 1

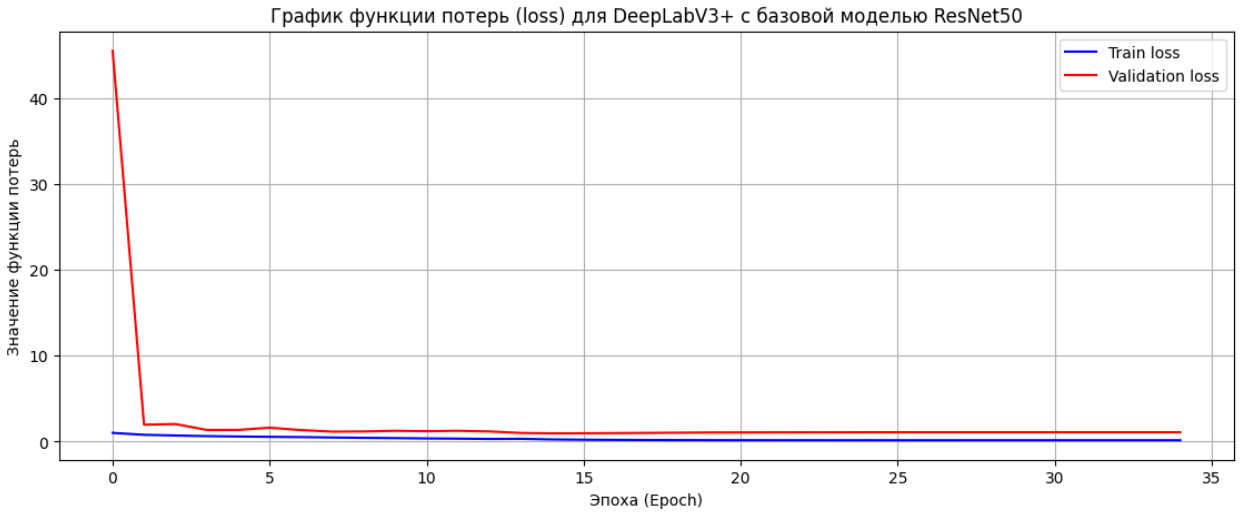


Рисунок 11 – График функции потерь для модели 2

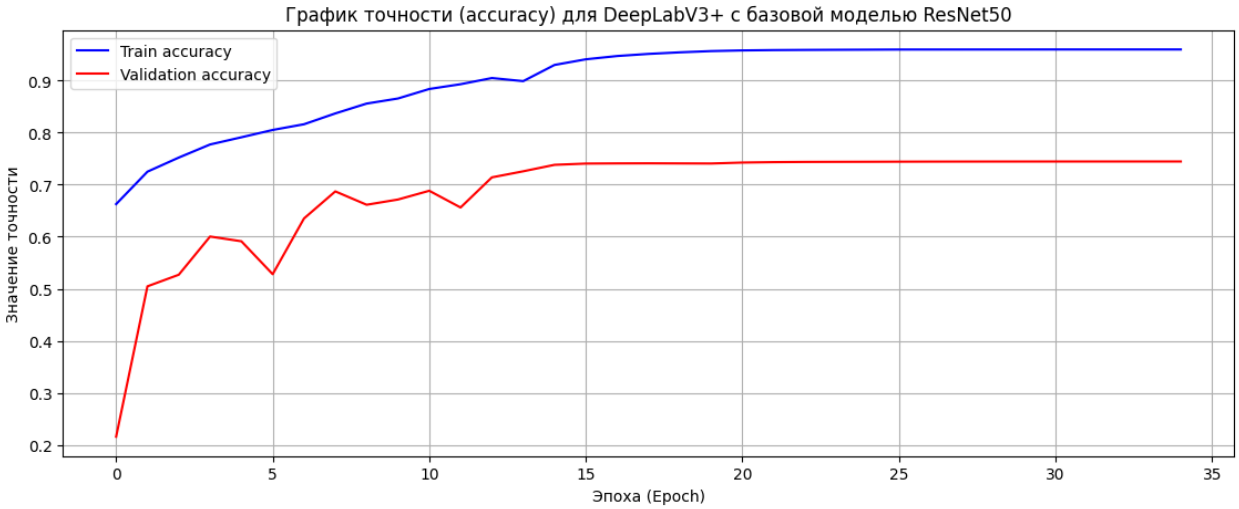
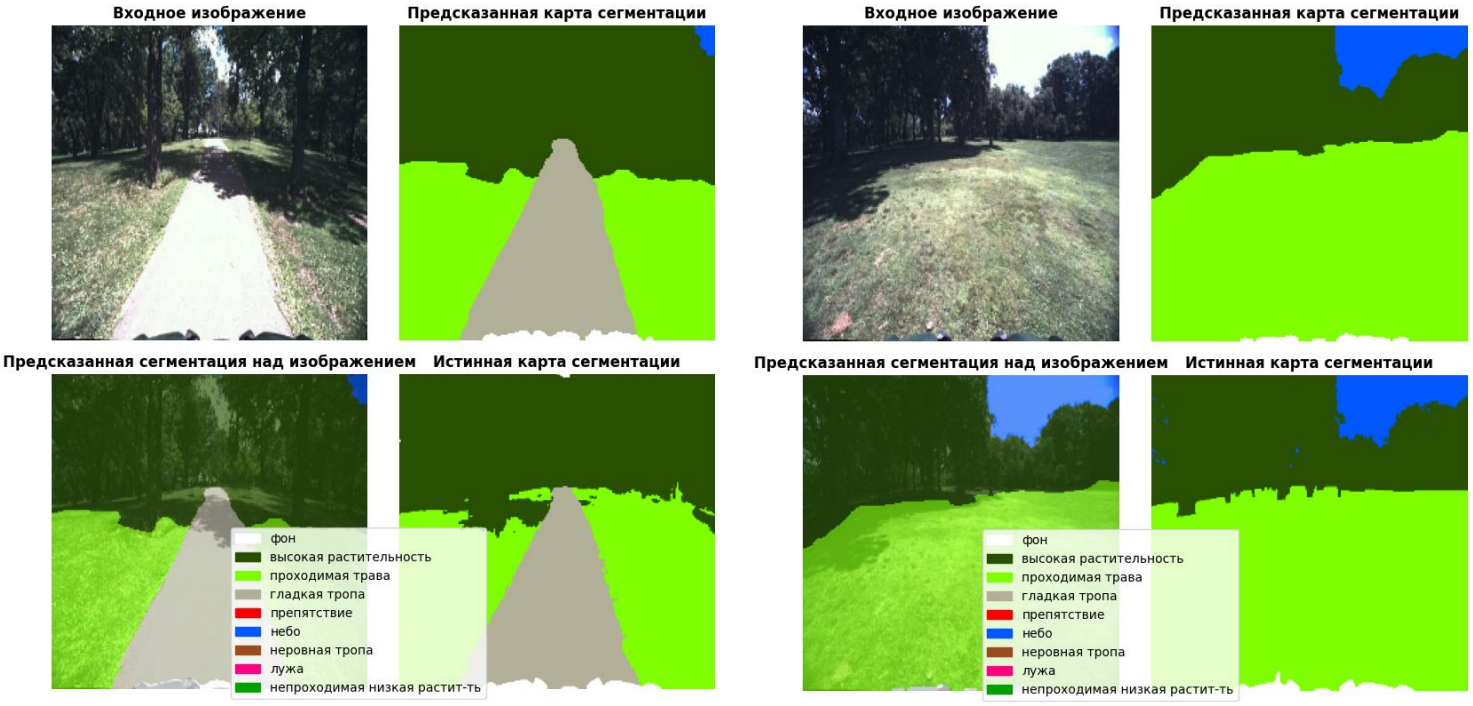


Рисунок 12 – График точности для модели 2

Качество сегментации

Качество моделей оценивалось по метрикам mIoU, Pixel Accuracy, Mean Accuracy на обучающей и валидационной выборках. Результаты приведены на рисунках 13 – 14 и в таблицах 3 – 4.



(a) (b)

Рисунок 13 – Результат применения модели 1 на изображении из обучающей (a) и валидационной (b) выборках. Исходное изображение, предсказанная маска сегментации, наложение предсказанной маски на исходное изображение, истинная маска сегментации

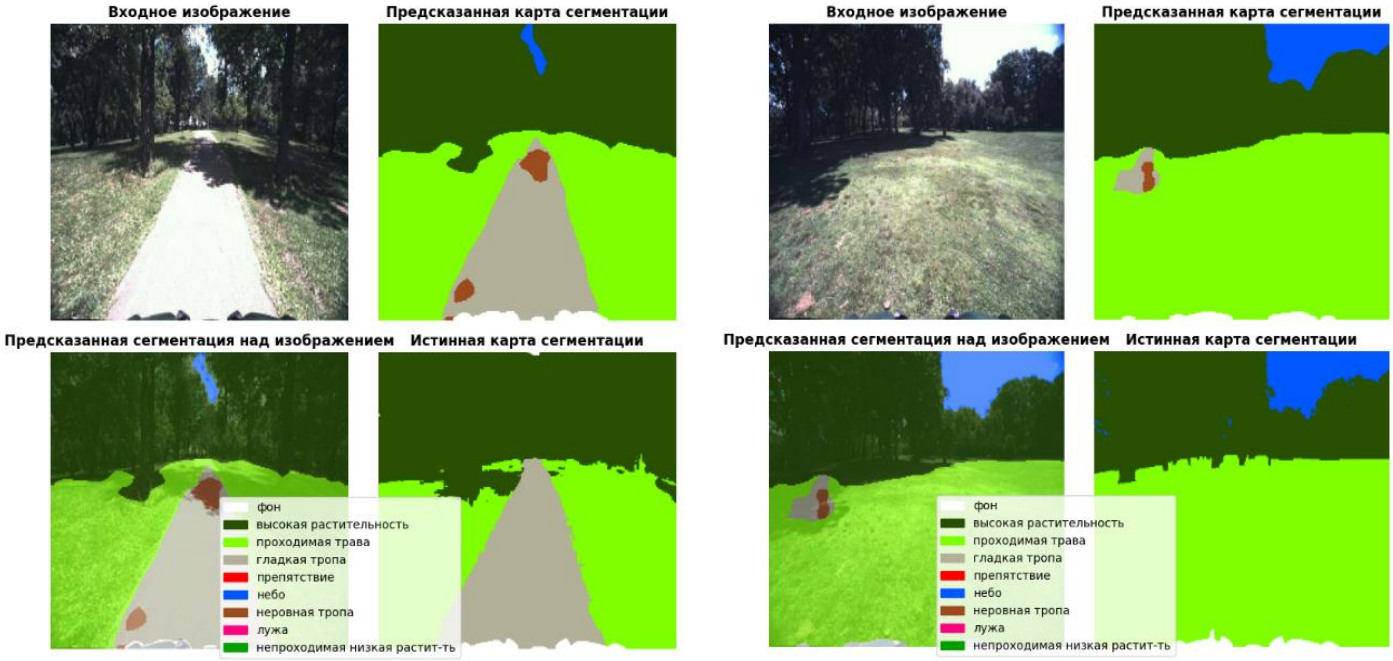


Рисунок 14 – Результат применения модели 2 на изображении из обучающей (a) и валидационной (b) выборках. Исходное изображение, предсказанная маска сегментации, наложение предсказанной маски на исходное изображение, истинная маска сегментации

Таблица 3

Качество моделей 1 и 2 на обучающей выборке

№	Метрика	Модель 1	Модель 2
1	mIoU	68.38%	56.38%
2	Pixel Accuracy	94.31%	93.95%
3	Mean Accuracy	98.74%	98.65%

Таблица 4

Качество моделей 1 и 2 на валидационной выборке

№	Метрика	Модель 1	Модель 2
1	mIoU	55.56%	49.30%
2	Pixel Accuracy	96.05%	94.55%
3	Mean Accuracy	99.12%	98.79%

Скорость обработки видеопоследовательностей

Производительность моделей оценивалась на основе числа кадров в секунду (FPS) при обработке видеопоследовательности, снятой в солнечный день, из которого был взят промежуток видео длительностью 1 минута 3 секунды. Пример работы, где на видео наложена предсказанная карта сегментации, изображен на рисунках 15 и 16.

Рисунок 15 – Пример сегментации сцены бездорожья при использовании модели 1 (на изображение из видео наложена предсказанная карта сегментации)

Рисунок 16 – Пример сегментации сцены бездорожья при использовании модели 2 (на изображение из видео наложена предсказанная карта сегментации)

На стенде 1 тестирование проводилось на ЦП и ГП (таблица 5), на стенде 2 – только на ЦП (таблица 6).

Таблица 5

Метрики числа кадров в секунду при обработке кадров видеопоследовательности с использованием моделей 1 и 2 на тестовом стенде 1 с использованием ГП и ЦП

№	Метрика	Модель 1		Модель 2	
		ГП	ЦП	ГП	ЦП
1	Среднее значение FPS	32.22	11.36	43.06	6.89
2	Медианное значение FPS	31	12	43	7
3	Минимальное значение FPS	1	0	0	1
4	Максимальное значение FPS	45	14	45	8
5	Стандартное отклонение FPS	3.52	1.14	1.35	0.48
6	10-й процентиль FPS	30	10	43	7
7	25-й процентиль FPS	30	11	43	7
8	50-й процентиль FPS	31	12	43	7
9	75-й процентиль FPS	32	12	43	7
10	90-й процентиль FPS	38	12	45	7

Таблица 6

Метрики числа кадров в секунду при обработке кадров видеопоследовательности с использованием моделей 1 и 2 на тестовом стенде 2 с использованием ЦП

№	Метрика	Модель 1	Модель 2
1	Среднее значение FPS	10.63	5.69
2	Медианное значение FPS	12	6
3	Минимальное значение FPS	0	0
4	Максимальное значение FPS	17	9
5	Стандартное отклонение FPS	3.95	1.8
6	10-й процентиль FPS	2	1
7	25-й процентиль FPS	10	6
8	50-й процентиль FPS	12	6
9	75-й процентиль FPS	12	7
10	90-й процентиль FPS	15	7

Сегментация трапециевидной области как области дорожного покрытия

Тестирование проводилось на видеопоследовательности, из которой был взят промежуток длительностью 14 секунд.

Имеются три региона интереса – регион 1, регион 2 и регион 3 – изображенные на рисунке 17.

Рисунок 17 – Область трапеции, соответствующая дороге, разбита на три региона интереса – 1, 2 и 3

На рисунках 18 – 23 приведены графики распределения классов по регионам 1, 2 и 3 трапеции выбранной видеопоследовательности для моделей 1 и 2.

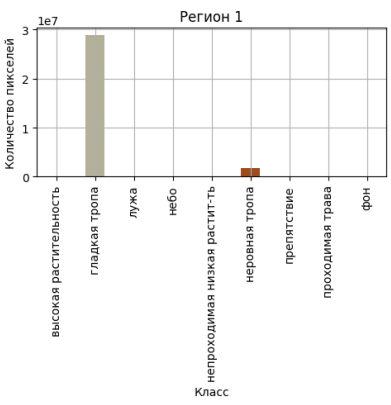


Рисунок 18 – Количество пикселей классов, соответствующих региону 1 трапеции при обработке кадров видеопоследовательности с использованием модели 1

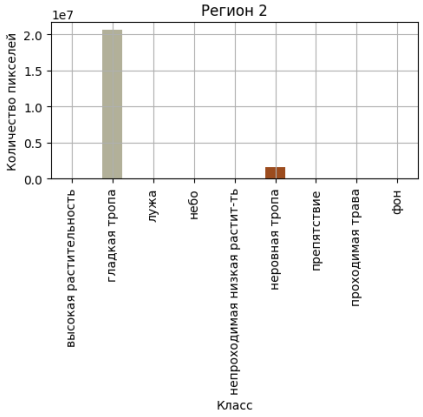


Рисунок 19 – Количество пикселей классов, соответствующих региону 2 трапеции при обработке кадров видеопоследовательности с использованием модели 1

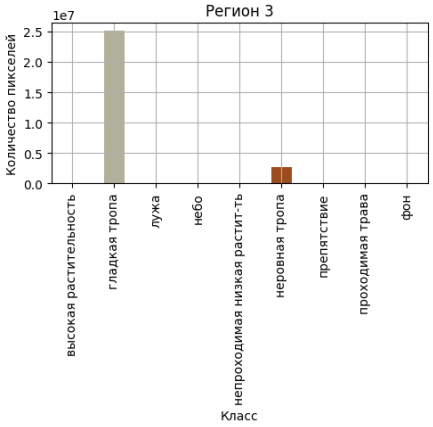


Рисунок 20 – Количество пикселей классов, соответствующих региону 3 трапеции при обработке кадров видеопоследовательности с использованием модели 1

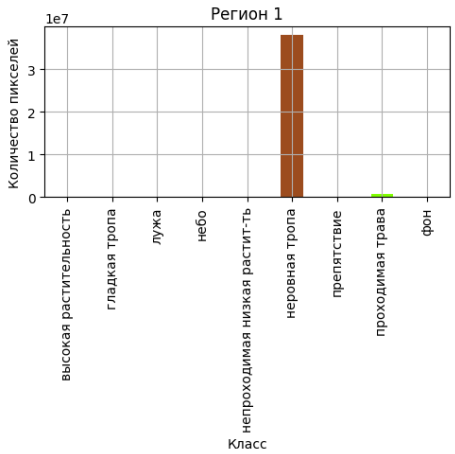


Рисунок 21 – Количество пикселей классов, соответствующих региону 1 трапеции при обработке кадров видеопоследовательности с использованием модели 2

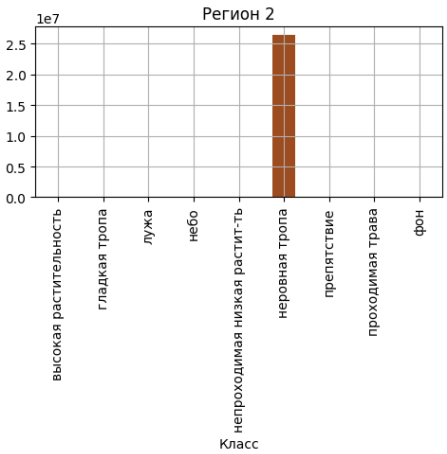


Рисунок 22 – Количество пикселей классов, соответствующих региону 2 трапеции при обработке кадров видеопоследовательности с использованием модели 2

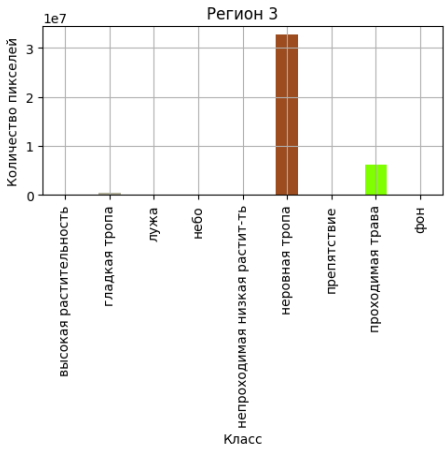


Рисунок 23 – Количество пикселей классов, соответствующих региону 3 трапеции при обработке кадров видеопоследовательности с использованием модели 2

На рисунках 24 и 25 представлены графики изменения распределения классов во времени для моделей 1 и 2.

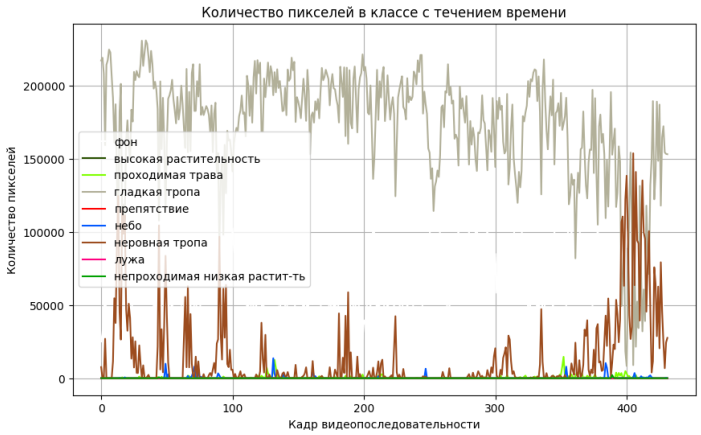


Рисунок 24 – Количество пикселей в классе с течением времени при обработке кадров видеопоследовательности с использованием модели 1

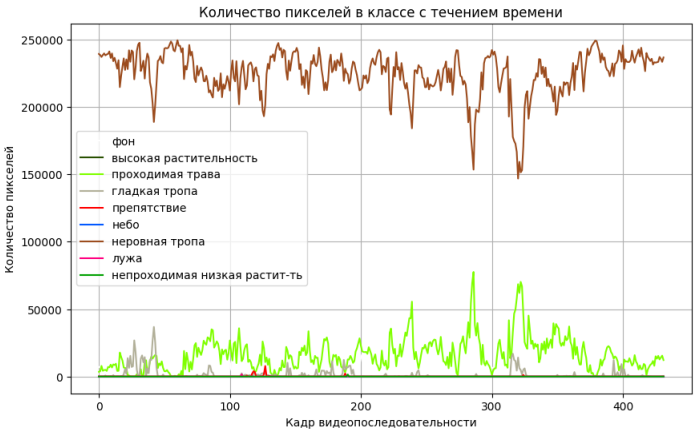


Рисунок 25 – Количество пикселей в классе с течением времени при обработке кадров видеопоследовательности с использованием модели 2

На рисунках 26, 27 представлены графики общего количества пикселей для всех классов объектов во всех регионах для моделей 1 и 2.

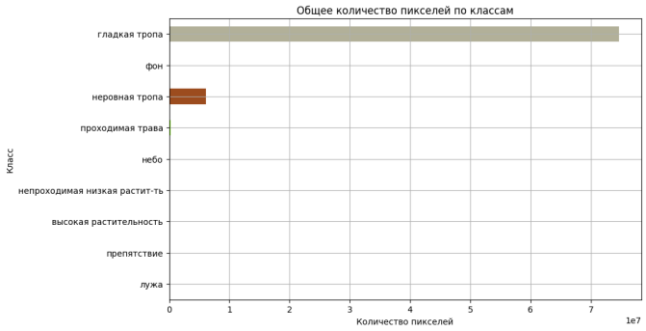


Рисунок 26 – Общее количество пикселей для всех классов объектов во всех регионах при обработке кадров видеопоследовательности с использованием модели 1

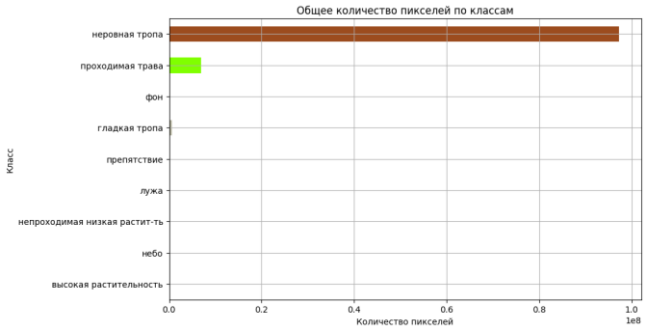


Рисунок 27 – Общее количество пикселей для всех классов объектов во всех регионах при обработке кадров видеопоследовательности с использованием модели 2

Сегментация и анализ средней линии трапециевидной области как ширины дорожного покрытия

Для тестирования использовалась видеопоследовательность, из которой был взят промежуток длительностью 14 секунд.

Для анализа бралась средняя линия трапеции, которая изображена линией красного цвета на рисунке 28.

Рисунок 28 – Средняя линия трапеции (горизонтальная линия красного цвета)

В обучающем наборе данных имеется несколько классов, которые можно использовать для движения робота. Это такие классы, как проходимая трава, гладкая тропа, неровная тропа. Так как для движения робота данные классы подходят, то была подсчитана длина наибольшей непрерывной линии, которая лежала на средней линии трапеции, состоящей из данных классов. График с

сравнением изменения ширины дороги в зависимости от кадра видеопоследовательности для моделей 1 и 2 изображен на рисунке 29.

Рисунок 29 – Сравнение изменения ширины дороги при обработке кадров видеопоследовательности с использованием моделей 1 и 2

В таблице 7 приведены метрики ширины дороги при выполнении сегментации выбранной видеопоследовательности.

Таблица 7

Метрики ширины дороги при обработке кадров видеопоследовательности с использованием моделей 1 и 2

№	Метрика	Модель 1 (значение)	Модель 2 (значение)
1	Среднее значение пикселей	751.74	768.95
2	Медианное значение пикселей	769.0	769.0
3	Минимальное значение пикселей	317	754
4	Максимальное значение пикселей	769	769
5	Стандартное отклонение пикселей	66.45	0.78
6	10-й процентиль пикселей	769.0	769.0
7	25-й процентиль пикселей	769.0	769.0
8	50-й процентиль пикселей	769.0	769.0
9	75-й процентиль пикселей	769.0	769.0
10	90-й процентиль пикселей	769.0	769.0

Устойчивость к изменяющимся условиям

Для оценки устойчивости проводилось тестирование, как модель ведет себя в различных изменяющихся условиях: изменения освещенности, скорости

движения, погодных условий на видеопоследовательностях с разрешением 1920×1080 пикселей и частотой 30 кадров в секунду.

Результаты сегментации моделей при различных условиях движения и освещенности (дорожная сцена, сцена бездорожья, раннее утро, сумерки, пасмурный день, зимняя сцена) и скоростях 5-16 км/ч приведены на рисунках 30 и 31.

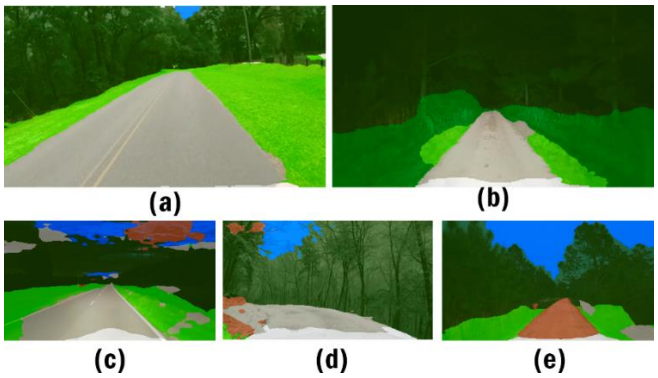


Рисунок 30 – Пример сегментации модели 1: (a) дорожная сцена (пасмурный день), (b) сцена бездорожья (раннее утро), (c) дорожная сцена (раннее утро), (d) зимняя дорожная сцена, (e) сцена бездорожья (сумерки) (на изображение из видео наложена предсказанная карта сегментации)

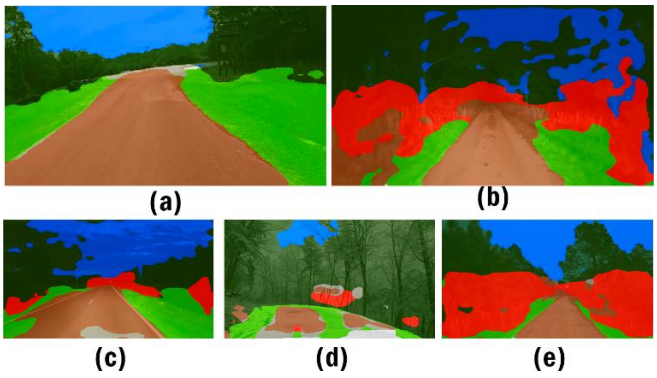


Рисунок 31 – Пример сегментации модели 2: (a) дорожная сцена (пасмурный день), (b) сцена бездорожья (раннее утро), (c) дорожная сцена (раннее утро), (d) зимняя дорожная сцена, (e) сцена бездорожья (сумерки) (на изображение из видео наложена предсказанная карта сегментации)

Обсуждение результатов

Сравнение качества сегментации

1. Сравнение метрик. Модель 1 показывает более высокие значения mIoU как на обучающей, так и на валидационной выборках – 68.38% против 56.38% и 55.56% против 49.30% соответственно. Модель 2 уступает по mIoU примерно на 12% на валидационной выборке, что указывает на меньшее качество обобщения.

2. Причины различий. Модель 1 использует в качестве базовой архитектуры MobileNetV3-Small, которая оптимизирована для работы на мобильных устройствах, поэтому более устойчива на ограниченном наборе данных, так как имеет меньшее число параметров, что снизит вероятность переобучения. Модель 2 имеет более глубокую и сложную архитектуру и включает в себя большее число параметров, что приведет к тому, что на

небольшом наборе данных будут проявляться переобучение и меньшая скорость сходимости к оптимальному результату.

Сравнение скорости обработки

Модели были протестированы на различных видеопоследовательностях со сценами бездорожья.

Анализ метрик, полученных при применении обученных нейронных сетей на тестовых стендах 1 и 2, показал (таблицы 5, 6):

1. При выполнении сегментации на графическом процессоре (тестовый стенд 1) модель 2 обеспечила большее количество кадров в секунду (FPS) в сравнении с моделью 1.
2. При выполнении сегментации на центральном процессоре модель 1 (стенды 1 и 2) обеспечила большее количество кадров в секунду (FPS) в сравнении с моделью 2.
3. Модель 1 продемонстрировала лучшие результаты по метрикам mIoU, Pixel Accuracy и Mean Accuracy на обучающей и валидационной выборках.

Таким образом, модель 1 имеет более высокие качество сегментации и скорость обработки изображений на центральном процессоре, модель 2 обладает более высокой скоростью обработки изображений на графическом процессоре.

Резюмируя вышесказанное:

1. Если вычислительные ресурсы ограничены и доступен только центральный процессор – рекомендуется использовать модель 1.
2. Если для работы доступен графический процессор, то рекомендуется использовать модель 2, так как достигается большее число кадров в секунду.

На тестовом стенде 1 обе модели обеспечили 30 кадров в секунду при выполнении семантической сегментации видеопоследовательностей с разрешением 1920×1080 пикселей на графическом процессоре – модель 1 с медианным значением 31 кадр в секунду и модель 2 с медианным значением 43 кадра в секунду.

Если источник видеопотока выдает 30 кадров в секунду, то при имеющихся вычислительных ресурсах, сопоставимых с тестовым стендом 1, обе модели

обеспечат семантическую сегментацию изображений с минимальной частотой 30 кадров в секунду.

При увеличении вычислительных ресурсов (более производительный графический процессор) скорость обработки кадров (сегментация изображений) возрастет.

Выбор между данными моделями зависит от решаемой задачи, где важны как точность, так и скорость сегментации при имеющихся вычислительных ресурсах.

Сравнение сегментации трапециевидной области как области дорожного покрытия

Обе архитектуры нейронных сетей (модель 1 и 2) продемонстрировали высокую точность сегментации трапециевидной области (области дорожного покрытия).

Модель 2 лучше выделяет участки дороги (доля преобладающего класса модели 2 90.08% против 69.17% у модели 1). Такие классы, как "лужа", "небо", "непроходимая низкая растит-ть", "препятствие", "проходимая трава", либо отсутствуют в регионах, либо количество их пикселей не превышает 1% от общего числа пикселей всех классов. Это свидетельствует о том, что внутри области трапеции преобладает дорога как объект.

Сегментация и анализ средней линии трапециевидной области как ширины дорожного покрытия

Сравнивая полученные результаты для двух архитектур нейронных сетей на основании данных, представленных в таблице 7, можно сделать вывод, что среднее значение ширины дороги у модели 2 выше, чем у модели 1. Медианное значение у обеих моделей одинаковое. Минимальное значение ширины дороги у модели 2 выше, чем у модели 1. Максимальное значение ширины дороги у обеих моделей одинаковое. Стандартное отклонение у модели 1 равно 66.45, из чего можно сделать вывод о значительной изменчивости ширины дороги. Стандартное отклонение модели 2 равно 0.78, что говорит о более стабильном результате. Процентили у обеих моделей равны 769, что свидетельствует о том, что большинство предсказанных результатов находятся в этом диапазоне.

Устойчивость к изменяющимся условиям

1. **Применимость к сценам, имеющим различные условия освещенности.** Обе архитектуры нейронных сетей справились с сегментацией объектов на видео с дорожными сценами и сценами бездорожья, записанными как в дневное, так и в ночное время (при наличии минимального искусственного освещения от фар транспортного средства), при различной яркости и условиях освещенности – солнечный день, пасмурный день, сумерки, искусственное освещение ночью.

2. **Применимость к сценам с различными временами года.** Для зимних сцен обе архитектуры нейронных сетей не продемонстрировали устойчивый результат. Это объясняется меньшим количеством изображений зимних сцен в наборе данных, на котором выполнялось обучение, по сравнению с другими временами года. Для других времен года получен удовлетворительный результат, явных аномалий, возникающих при сегментации объектов на видео, не выявлено.

3. **Применимость для различных скоростей движения транспорта.** Обе архитектуры нейронных сетей справились с сегментации объектов на видео с частотой 30 кадров в секунду, записанными при движении транспортного средства со скоростями 5, 10, 16, 30 км/ч.

Заключение

В ходе выполнения работы проведено исследование двух композитных архитектур нейронных сетей на основе DeepLabv3+ с использованием модифицированных кодировщиков MobileNetV3-Small и ResNet50 для решения задачи семантической сегментации изображений, предназначенные для использования в системах, работающих в реальном времени.

В качестве основы использовались веса моделей MobileNetV3-Small и ResNet50, обученных на наборе данных ImageNet. Затем выполнено дообучение двух разработанных архитектур нейронных сетей на подготовленном наборе данных YCOR.

Проведен сравнительный анализ разработанных моделей, который показал следующие результаты:

1. **Качество сегментации.** DeepLabv3+ с базовой моделью MobileNetV3-Small продемонстрировала более высокое качество сегментации на обучающей и валидационной выборках (метрики mIoU, Pixel Accuracy, Mean Accuracy) по сравнению с DeepLabv3+ с базовой моделью ResNet50.

2. **Скорость обработки.** С учетом ожидаемых скоростей движения робота от 5 до 40 км/ч требовалась обработка от 15 до 30 кадров в секунду, в зависимости от скорости движения. При выполнении сегментации на графическом процессоре обе модели обеспечили 30 кадров в секунду при выполнении семантической сегментации видеопоследовательностей с разрешением 1920×1080 пикселей на графическом процессоре – модель 1 с медианным значением 31 кадр в секунду и модель 2 с медианным значением 43 кадра в секунду. При сегментации на центральном процессоре скорость обработки ниже, при этом модель 1 превзошла модель 2 по скорости.

3. **Сегментация трапециевидной области как области дорожного покрытия.** Обе архитектуры нейронных сетей продемонстрировали высокую точность сегментации трапециевидной области. Модель 2 лучше выделяет участки дороги (доля преобладающего класса модели 2 90.08% против 69.17% у модели 1).

4. **Сегментация средней линии трапециевидной области как ширины дорожного покрытия.** Средняя ширина дороги у модели 2 выше, распределение значений более стабильное (стандартное отклонение 0.78 против 66.45 у модели 2), что указывает на меньшую изменчивость ширины дороги.

5. **Устойчивость к изменяющимся условиям.** Обе модели корректно сегментируют объекты при различной освещенности, скоростях движения (5, 10, 16, 30 км/ч) и погодных условиях, за исключением зимних сцен, для которых количество обучающих изображений представлено меньше по сравнению с другими временами года.

Конфликт интересов

Автор заявляет об отсутствии конфликта интересов.

Conflict of interest

The author declares no conflict of interest.

Список источников

1. Олькина Д.С. Алгоритм семантической сегментации изображений для решения задачи позиционирования летательного аппарата на земной поверхности // Труды МАИ. 2023. № 130. DOI: 10.34759/trd-2023-130-18
2. Тонких А.Н. Применение нейросетевых технологий для распознавания распределенных объектов на радиолокационных изображениях // Труды МАИ. 2025. № 141. URL: https://trudymai.ru/published.php?ID=184504
3. Митькин М.А., Гаврилов К.Ю. Применение искусственных нейронных сетей для восстановления объектов на радиолокационных изображениях // Труды МАИ. 2025. № 141. URL: https://trudymai.ru/published.php?ID=184505
4. Компьютерное зрение [Электронный ресурс] / Л.Шапиро, Дж. Стокман ; пер. с англ. 2-е изд. (эл.). М. : БИНОМ. Лаборатория знаний, 2013. 752 с. : ил.
5. Image Thresholding // OpenCV URL: https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html (дата обращения: 17.02.2026).
6. Canny Edge Detection // OpenCV URL: https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html (дата обращения: 17.02.2026).
7. J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015 pp. 3431-3440. doi: 10.1109/CVPR.2015.7298965
8. H. Noh, S. Hong and B. Han, "Learning Deconvolution Network for Semantic Segmentation," in 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015 pp. 1520-1528. doi: 10.1109/ICCV.2015.178

9. Badrinarayanan, Vijay & Kendall, Alex & Cipolla, Roberto. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. doi: https://doi.org/10.17863/CAM.17966

10. Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science(), vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28

11. Chen, Liang-Chieh & Papandreou, George & Kokkinos, Iasonas & Murphy, Kevin & Yuille, Alan. (2015). Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs.

12. Chen, Liang-Chieh & Papandreou, George & Kokkinos, Iasonas & Murphy, Kevin & Yuille, Alan. (2016). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence. PP. 10.1109/TPAMI.2017.2699184.

13. Chen, Liang-Chieh & Papandreou, George & Schroff, Florian & Adam, Hartwig. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation.

14. Chen, LC., Zhu, Y., Papandreou, G., Schroff, F., Adam, H. (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science(), vol 11211. Springer, Cham. https://doi.org/10.1007/978-3-030-01234-2_49

15. P. N. Hadinata, D. Simanta, L. Eddy, and K. Nagai, "Crack Detection on Concrete Surfaces Using Deep Encoder-Decoder Convolutional Neural Network: A Comparison Study Between U-Net and DeepLabV3+," Journal of the Civil Engineering Forum, vol. 7, no. 3, p. 323, Aug. 2021, doi: https://doi.org/10.22146/jcef.65288.

16. A. Howard and M. Sandler and B. Chen and W. Wang and L. Chen and M. Tan and G. Chu and V. Vasudevan and Y. Zhu and R. Pang and H. Adam and Q. Le Searching for MobileNetV3 // 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Los Alamitos, CA, USA: IEEE Computer Society, 2019. C. 1314-1324.

17. He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian Deep Residual Learning for Image Recognition // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. С. 770-778.

18. M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” arXiv.org, Sep. 11, 2020. http://arxiv.org/abs/1905.11946

19. M. Tan and Q. V. Le, “EfficientNetV2: Smaller Models and Faster Training,” arxiv.org, Apr. 2021, doi: https://doi.org/10.48550/arXiv.2104.00298.

20. T. Shahriar, “Comparative Analysis of Lightweight Deep Learning Models for Memory-Constrained Devices,” arXiv.org, 2025. https://arxiv.org/abs/2505.03303v2

21. S. Hu, J. Liu, and Z. Kang, “DeepLabV3+/Efficientnet Hybrid Network-Based Scene Area Judgment for the Mars Unmanned Vehicle System,” Sensors (Basel, Switzerland), vol. 21, no. 23, p. 8136, Dec. 2021, doi: https://doi.org/10.3390/s21238136.

22. Maturana, Daniel and Chou, Po-Wei and Uenoyama, Masashi and Scherer, Sebastian Real-time semantic mapping for autonomous off-road navigation // Field and Service Robotics. 2018. С. 335-350.

23. Yamaha-CMU Off-Road Dataset Converter to ADE20K Format // Github URL: https://gist.github.com/GerardMaggiolino/258a65077d43d4e176e0fb0240a49edb (дата обращения: 03.03.2025).

References

1. Ol'kina D.S. Algoritm semanticheskoi segmentatsii izobrazhenii dlya resheniya zadachi pozitsionirovaniya letatel'nogo apparata na zemnoi poverkhnosti // Trudy MAI. 2023. № 130. DOI: 10.34759/trd-2023-130-18

2. Tonkikh A.N. Primenenie neirosetevykh tekhnologii dlya raspoznavaniya raspredelennykh ob"ektov na radiolokatsionnykh izobrazheniyakh // Trudy MAI. 2025. № 141. URL: https://trudymai.ru/published.php?ID=184504

3. Mit'kin M.A., Gavrilov K.Yu. Primenenie iskusstvennykh neironnykh setei dlya vosstanovleniya ob"ektov na radiolokatsionnykh izobrazheniyakh // Trudy MAI. 2025. № 141. URL: https://trudymai.ru/published.php?ID=184505

4. Komp'yuternoe zrenie [Elektronnyi resurs] / L.Shapiro, Dzh. Stokman ; per. s angl. 2-e izd. (el.). M. : BINOM. Laboratoriya znanii, 2013. 752 s. : il.

5. Image Thresholding // OpenCV URL: https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html (accessed: 17.02.2026).

6. Canny Edge Detection // OpenCV URL: https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html (accessed: 17.02.2026).

7. J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015 pp. 3431-3440. doi: 10.1109/CVPR.2015.7298965

8. H. Noh, S. Hong and B. Han, "Learning Deconvolution Network for Semantic Segmentation," in 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015 pp. 1520-1528. doi: 10.1109/ICCV.2015.178

9. Badrinarayanan, Vijay & Kendall, Alex & Cipolla, Roberto. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. doi: https://doi.org/10.17863/CAM.17966

10. Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science(), vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28

11. Chen, Liang-Chieh & Papandreou, George & Kokkinos, Iasonas & Murphy, Kevin & Yuille, Alan. (2015). Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs.

12. Chen, Liang-Chieh & Papandreou, George & Kokkinos, Iasonas & Murphy, Kevin & Yuille, Alan. (2016). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence. PP. 10.1109/TPAMI.2017.2699184.

13. Chen, Liang-Chieh & Papandreou, George & Schroff, Florian & Adam, Hartwig. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation.

14. Chen, LC., Zhu, Y., Papandreou, G., Schroff, F., Adam, H. (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science(), vol 11211. Springer, Cham. https://doi.org/10.1007/978-3-030-01234-2_49

15. P. N. Hadinata, D. Simanta, L. Eddy, and K. Nagai, "Crack Detection on Concrete Surfaces Using Deep Encoder-Decoder Convolutional Neural Network: A Comparison Study Between U-Net and DeepLabV3+," Journal of the Civil Engineering Forum, vol. 7, no. 3, p. 323, Aug. 2021, doi: https://doi.org/10.22146/jcef.65288.

16. A. Howard and M. Sandler and B. Chen and W. Wang and L. Chen and M. Tan and G. Chu and V. Vasudevan and Y. Zhu and R. Pang and H. Adam and Q. Le Searching for MobileNetV3 // 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Los Alamitos, CA, USA: IEEE Computer Society, 2019. C. 1314-1324.

17. He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian Deep Residual Learning for Image Recognition // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. C. 770-778.

18. M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," arXiv.org, Sep. 11, 2020. http://arxiv.org/abs/1905.11946

19. M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," arxiv.org, Apr. 2021, doi: https://doi.org/10.48550/arXiv.2104.00298.

20. T. Shahriar, "Comparative Analysis of Lightweight Deep Learning Models for Memory-Constrained Devices," arXiv.org, 2025. https://arxiv.org/abs/2505.03303v2

21. S. Hu, J. Liu, and Z. Kang, "DeepLabV3+/Efficientnet Hybrid Network-Based Scene Area Judgment for the Mars Unmanned Vehicle System," Sensors (Basel, Switzerland), vol. 21, no. 23, p. 8136, Dec. 2021, doi: https://doi.org/10.3390/s21238136.

22. Maturana, Daniel and Chou, Po-Wei and Uenoyama, Masashi and Scherer, Sebastian Real-time semantic mapping for autonomous off-road navigation // Field and Service Robotics. 2018. C. 335-350.

23. Yamaha-CMU Off-Road Dataset Converter to ADE20K Format // Github URL: https://gist.github.com/GerardMaggiolino/258a65077d43d4e176e0fb0240a49edb (accessed: 03.03.2025).

Информация об авторах

Николай Георгиевич Корыткин, аспирант, Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет имени М.В. Ломоносова», г. Москва, Россия; ORCID: https://orcid.org/0000-0002-6685-7029; e-mail: korytkinng@my.msu.ru

Information about the authors

Nikolai G. Korytkin, Postgraduate Student; Federal State Budget Educational Institution of Higher Education M.V. Lomonosov Moscow State University, Moscow, Russia; ORCID: https://orcid.org/0000-0002-6685-7029;
e-mail: korytkinng@my.msu.ru

Получено 30 ноября 2025 ● Принято к публикации 19 февраля 2026 ● Опубликовано 27 февраля 2026
Received 30 November 2025 ● Accepted 19 February 2026 ● Published 27 February 2026