Применение сверточных нейронных сетей (CNN) для сегментации изображений и обнаружения проходимых поверхностей в задачах робототехники и беспилотных систем

27 августа 2023 г.

Применение сверточных нейронных сетей (CNN) для сегментации изображений и обнаружения проходимых поверхностей в задачах робототехники и беспилотных систем

0.1. Введение

В современной робототехнике важную роль играет точное понимание окружающей среды и обеспечение безопасной навигации. Одним из инструментов для решения этой задачи являются сверточные нейронные сети, демонстрирующие впечатляющие результаты в обработке и анализе изображений.

0.2. Цели и задачи

Целью данной работы является сравнение эффективности различных архитектур сверточных нейронных сетей (CNN) в задаче сегментации изображений для обнаружения проходимых поверхностей в робототехнике и беспилотных системах. Для достижения этой цели выделяются следующие задачи: 1. Изучение предметной области и обзор существующих методов сегментации изображений. 2. Выбор наборов данных, подходящих для проведения экспериментов с различными архитектурами CNN. 3. Реализация и обучение различных архитектур CNN, включая U-Net, FCN, DeepLab и другие, на выбранных наборах данных. Задача включает определение функций потерь и оптимизаторов. 4. Вычисление метрик производительности для каждой архитектуры, таких как точность, полнота, F1-мера и другие, на тестовых данных. Эта задача позволит провести объективное сравнение архитектур и определить наиболее эффективные. 5. Анализ полученных результатов и интерпретация производительности каждой архитектуры. 6. Формулирование выводов о наиболее подходящей архитектуре CNN для задачи сегментации изображений в контексте обнаружения проходимых поверхностей. Данная задача даст возможность сделать практические рекомендации для применения полученных результатов в робототехнике и беспилотных системах.

0.3. Изучение предметной области

Существует больше количество методов сегментации изображений:

1. **Методы на основе кластеризации:** Эти методы группируют пиксели или регионы изображения в кластеры на основе их сходства, позволяя разделить изображение на различные сегменты по характеристикам, таким как цвет или текстура.
2. **Сегментация с использованием гистограммы:** Этот метод основан на анализе гистограммы яркости или цветовых компонент изображения. Пороги применяются для разделения изображения на сегменты с разной интенсивностью или цветом.
3. **Методы выделения краёв:** Эти методы сосредотачиваются на выделении перепадов интенсивности или текстурных характеристик на изображении, что позволяет определить границы объектов и разделить изображение на сегменты.
4. **Суперпиксельная сегментация:** Данный метод разбивает изображение на компактные регионы (суперпиксели), обычно схожие по цвету или текстуре, что упрощает дальнейший анализ и сегментацию.
5. **Сегментация методом водораздела:** Этот метод сегментации опирается на градиентные характеристики изображения. Он моделирует изображение как граф, где пиксели представлены вершинами. Пиксели, имеющие наибольшую абсолютную величину

градиента яркости, соответствуют линиям водораздела, которые представляют границы областей.

6. **Сегментация с помощью моделей:** Сегментация на основе модели предполагает использование статистических моделей или заранее определенных шаблонов для идентификации и изоляции определенных областей или объектов на изображении. Эти модели отражают характеристики интересующих объектов, такие как форма, текстура или распределение цвета. Процесс сегментации направлен на различение компонентов или регионов на основе заранее определенных критериев, установленных моделью. Этот подход особенно полезен, когда есть предварительные знания об ожидаемом внешнем виде или свойствах сегментируемых объектов.
7. **Обучаемая сегментация**: Данный метод основан на использовании обучающих данных для создания модели, способной автоматически выделять объекты и регионы на изображении. С использованием алгоритмов машинного обучения, таких как нейронные сети, модель обучается распознавать и классифицировать разные части изображения. Это позволяет обнаруживать объекты с высокой точностью, даже в условиях изменяющейся окружающей среды и сложных сценариях. Данный метод обучаемой сегментации имеет преимущество адаптации к различным типам данных и способности улучшать свою производительность с накоплением опыта. Однако он также требует большого объема размеченных данных для обучения, и правильный выбор архитектуры нейронной сети играет решающую роль в достижении высокой точности сегментации.

Эти методы предоставляют разнообразные способы сегментации изображений, каждый из которых имеет свои преимущества и ограничения в зависимости от характеристик задачи и типа данных.

Основные методы и подходы к решению задачи сегментации изображений с использованием сверточных нейронных сетей (CNN) включают: 1. **U-Net:** Архитектура U-Net состоит из энкодера, который извлекает признаки из изображения, и декодера, который восстанавливает пространственное разрешение изображения. Этот метод обеспечивает точное восстановление границ и деталей объектов на изображении. 2. **Fully Convolutional Network (FCN):** FCN использует полносверточную архитектуру, где сверточные слои используются для обработки всего изображения без уплотнения. 3. **DeepLab:** Архитектура DeepLab базируется на использовании дополнительных модулей для учета контекста изображения, таких как асимметричные сверточные фильтры или деконволюционные слои. 4. **Mask R-CNN:** Mask R-CNN расширяет архитектуру Faster R-CNN для сегментации объектов путем предсказания масок пиксельных уровней для каждого объекта. Это позволяет точно выделить границы и области объектов на изображении. 5. **Feature Pyramid Network (FPN) :** Основной концепцией FPN является построение ``пирамиды'' из признаковых карт разного масштаба, что позволяет модели эффективно улавливать детали и контекст изображения разных размеров. FPN строит пирамидальную структуру путем соединения признаков с разных уровней. Более низкие уровни (с более высоким разрешением) пропускаются вверх через операции пулинга или простой интерполяции, чтобы получить более высокую абстракцию на более низких уровнях.

Это лишь некоторые из множества архитектур, используемых для семантической сегментации изображений.

0.4. Определение входных данных, выходных данных и метрик для оценки производительности

Для проведения исследования, следует определить характеристики входных данных, ожидаемые выходные данные и метрики для оценки производительности.

0.4.1. Входные данные

Набор данных должен включает изображения, представляющие различные сцены и условия, чтобы модель могла обобщать и правильно сегментировать объекты в разных ситуациях. Входные данные должны быть предварительно обработаны и нормализованы. Каждое изображение сопровождается соответствующей разметкой, где каждый пиксель объекта или интересующей области имеет свой соответствующий класс.

0.4.2. Выходные данные

Выходные данные - это сегментированные карты, где каждый пиксель обозначает класс объекта или области. Таким образом, для каждого пикселя на выходе сети ожидается соответствующий класс (например, объект или фон).

0.4.3. Метрики для оценки производительности

Для оценки производительности моделей необходимо использовать соответствующие метрики. В данной работе можно использовать следующие метрики: - IoU (Intersection over Union): - Dice coefficient - Accuracy - Precision и Recall

IoU (Intersection over Union) Измеряет степень перекрытия между предсказанными и истинными сегментами объектов. IoU (Intersection over Union) - это метрика, используемая для оценки качества сегментации в задачах компьютерного зрения, включая семантическую сегментацию. Она измеряет степень перекрытия между областью, предсказанной моделью, и истинной разметкой.

Формула для вычисления метрики IoU выглядит следующим образом:

$IoU = \frac{TP}{TP+FP+FN}$

Где: - (TP) (True Positive) - количество пикселей, которые правильно классифицированы как объект на изображении. - (FP) (False Positive) - количество пикселей, которые неправильно классифицированы как объект на изображении (ложные срабатывания). - (FN) (False Negative) - количество пикселей, которые неправильно классифицированы как фон на изображении (пропущенные объекты).

Значение метрики IoU варьируется от 0 до 1, где значение 0 означает полное несовпадение между предсказанными и истинными разметками, а значение 1 означает полное совпадение.

IoU позволяет оценить, насколько точно модель сегментирует объекты на изображении и насколько хорошо она выделяет их границы. Эта метрика является одной из наиболее распространенных и информативных метрик для оценки качества сегментации.

Dice coefficient Dice coefficient (также известный как F1-score) - это еще одна метрика, используемая для оценки качества сегментации в задачах компьютерного зрения. Он измеряет сходство между предсказанными сегментированными областями и истинной разметкой.

Формула для вычисления метрики Dice coefficient выглядит следующим образом:

$Dice = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$

Где: - (TP) (True Positive) - количество пикселей, которые правильно классифицированы как объект на изображении. - (FP) (False Positive) - количество пикселей, которые неправильно классифицированы как объект на изображении (ложные срабатывания). - (FN) (False Negative) - количество пикселей, которые неправильно классифицированы как фон на изображении (пропущенные объекты).

Аналогично метрике IoU, значение метрики Dice coefficient также варьируется от 0 до 1, где 0 означает полное несовпадение, а 1 - полное совпадение.

Метрика Dice coefficient подчеркивает сходство между предсказанными и истинными сегментированными областями, а также позволяет оценить эффективность модели в выделении объектов и их границ на изображении. Эта метрика особенно полезна, когда требуется учитывать баланс между точностью и полнотой в оценке качества сегментации.

Accuracy Accuracy - это метрика, используемая для оценки общей правильности классификации или сегментации модели. Она измеряет долю правильно классифицированных пикселей (или объектов) от общего числа пикселей (или объектов) на изображении.

Формула для вычисления метрики Accuracy выглядит следующим образом:

$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$

Где: - (TP) (True Positive) - количество пикселей, которые правильно классифицированы как объект на изображении. - (TN) (True Negative) - количество пикселей, которые правильно классифицированы как фон на изображении. - (FP) (False Positive) - количество пикселей, которые неправильно классифицированы как объект на изображении (ложные срабатывания). - (FN) (False Negative) - количество пикселей, которые неправильно классифицированы как фон на изображении (пропущенные объекты).

Значение метрики Accuracy также варьируется от 0 до 1.

Accuracy позволяет оценить общую производительность модели, но она может быть недостаточно информативной в случае, когда классы на изображении несбалансированы, и один класс преобладает над другим.

Precision (точность) Эта метрика измеряет долю правильно классифицированных положительных объектов (истинно положительных) относительно всех объектов, которые модель классифицировала как положительные. Precision оценивает, насколько модель правильно идентифицирует объекты, избегая ложных срабатываний.

Формула для вычисления метрики Precision выглядит следующим образом:

$Precision = \frac{TP}{TP + FP}$

Где: - (TP) (True Positive) - количество объектов, которые правильно классифицированы как положительные. - (FP) (False Positive) - количество объектов, которые неправильно классифицированы как положительные (ложные срабатывания).

Recall (полнота) Recall измеряет долю правильно классифицированных положительных объектов (истинно положительных) относительно всех истинно положительных объектов в данных. Recall оценивает, насколько модель успешно находит все положительные объекты.

Формула для вычисления метрики Recall выглядит следующим образом:

$Recall = \frac{TP}{TP+FN}$

Где: - (TP) (True Positive) - количество объектов, которые правильно классифицированы как положительные. - (FN) (False Negative) - количество объектов, которые неправильно классифицированы как отрицательные (пропущенные положительные объекты).

Precision и Recall тесно связаны между собой и представляют компромисс между точностью и полнотой классификации. Выбор между этими метриками зависит от приоритетов задачи. Высокое значение Precision указывает на мало ложных срабатываний, в то время как высокое значение Recall показывает, что модель обнаруживает большую долю положительных объектов.

0.5. Выбор и предобработка данных

Для проведения исследования был выбран датасет ADE20K, предоставляющий обширную коллекцию изображений разнообразных сцен с соответствующей семантической разметкой.

Предобработка данных

1. **Нормализация:** Изображения из датасета подвергаются нормализации, чтобы значения пикселей находились в диапазоне [0, 1]. Это помогает улучшить стабильность обучения моделей.
2. **Размер изображений:** Изображения изменены до фиксированного разрешения 256x256, для обеспечения единообразия и увеличения скорости обучения.
3. **Расширение датасета:** Для увеличения разнообразия и обобщающей способности моделей на входных данных проводится аугментация. Она включает случайные повороты, зеркальные отражения, что помогает создать разнообразные варианты изображений.
4. **Преобразование разметки:** Семантическая разметка изображений в датасете предоставляется в виде масок с числовыми идентификаторами классов. Эти маски преобразуются в бинарные маски (One-hot encoding), где каждый пиксель относится к объекту (1) или фону (0).
5. **Разделение на обучение и валидацию:** Данный датасет уже разделен на обучающую (20210 изображений) и валидационную (2000 изображений) выборки. Процесс обучение сети производится на обучающей выборке, а валидационная выборка используется для оценки производительности моделей.

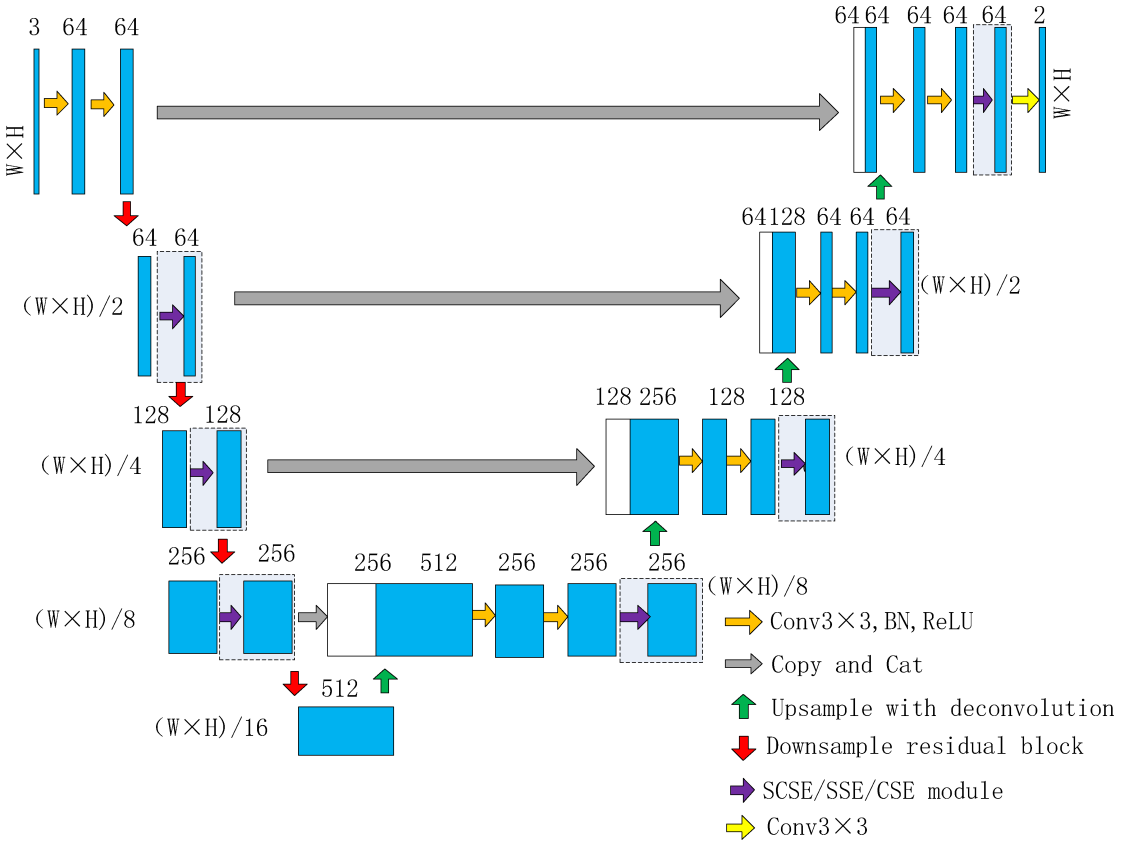
0.6. Архитектуры сверточных нейронных сетей (CNN)

0.6.1. U-Net

U-Net - это глубокая архитектура сверточной нейронной сети, разработанная для сегментации биомедицинских изображений. Её особенностью является соединение путей кодирования и декодирования.

Основные характеристики:

1. **Кодировщик (Encoder):** В начале архитектуры располагается путь свертки, включающий серию сверточных и пулинговых слоев. Этот путь служит для постепенного уменьшения размера изображения и извлечения высокоуровневых признаков.
2. **Декодировщик (Decoder):** Путь декодирования состоит из серии сверточных слоев, каждый из которых увеличивает размер изображения. Слой декодирования соединен с соответствующим слоем пути свертки с тем же масштабом.
3. **Сквозные связи:** Одной из ключевых особенностей U-Net являются сквозные связи, которые передают информацию между путями свертки и декодирования. Это позволяет сети одновременно извлекать детали объектов и удерживать глобальный контекст.
4. **Функции активации:** В U-Net используются функции активации, такие как ReLU (Rectified Linear Activation), для активации скрытых слоев и внесения нелинейности.

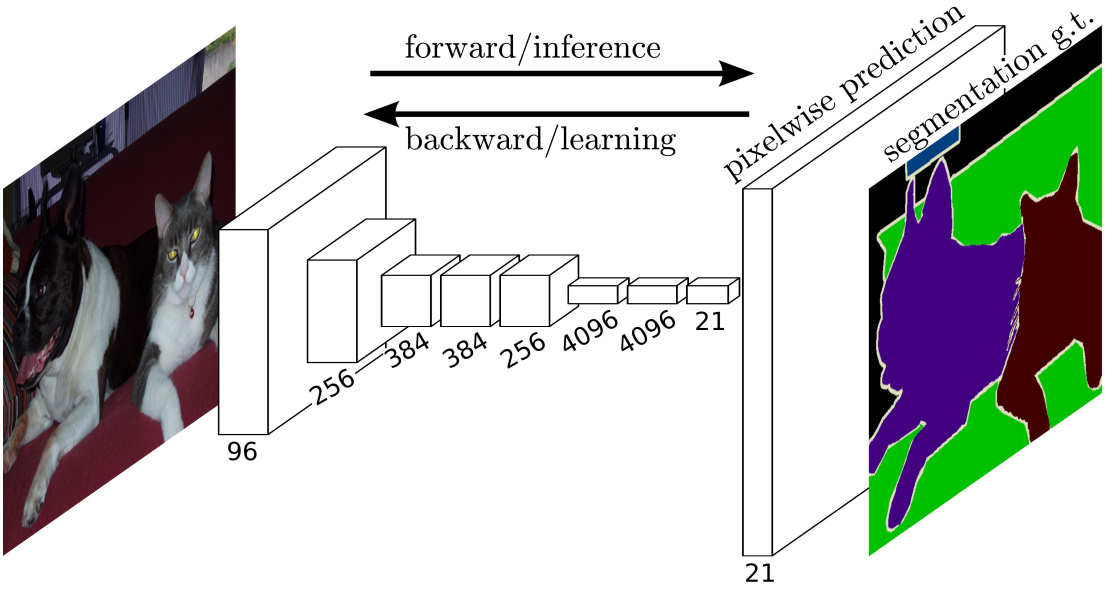


0.6.2. FCN (Fully Convolutional Network)

FCN - это сверточная нейронная сеть, разработанная специально для задачи семантической сегментации изображений.

Основные характеристики:

1. **Сверточные слои:** Весь путь архитектуры FCN состоит из сверточных слоев. Отсутствие полносвязанных слоев позволяет работать с изображениями разных размеров и выполнять операции на пиксельном уровне.
2. **Сквозные соединения (Skip Connections):** Чтобы учесть детали объектов на разных масштабах, FCN использует сквозные соединения. Эти соединения передают информацию из слоев пути свертки к соответствующим слоям декодирования.
3. **Апсемплинг:** Для восстановления пространственной информации после сверточных операций уменьшения размера, в FCN используются операции апсемплинга. Обратные операции пулинга (unpooling) или операции деконволюции используются для восстановления размера изображения.
4. **Softmax слой:** В конце архитектуры применяется Softmax слой, который представляет вероятности для каждого пикселя относительно различных классов. Это позволяет получить карту сегментации, на которой каждому пикселю присваивается класс.



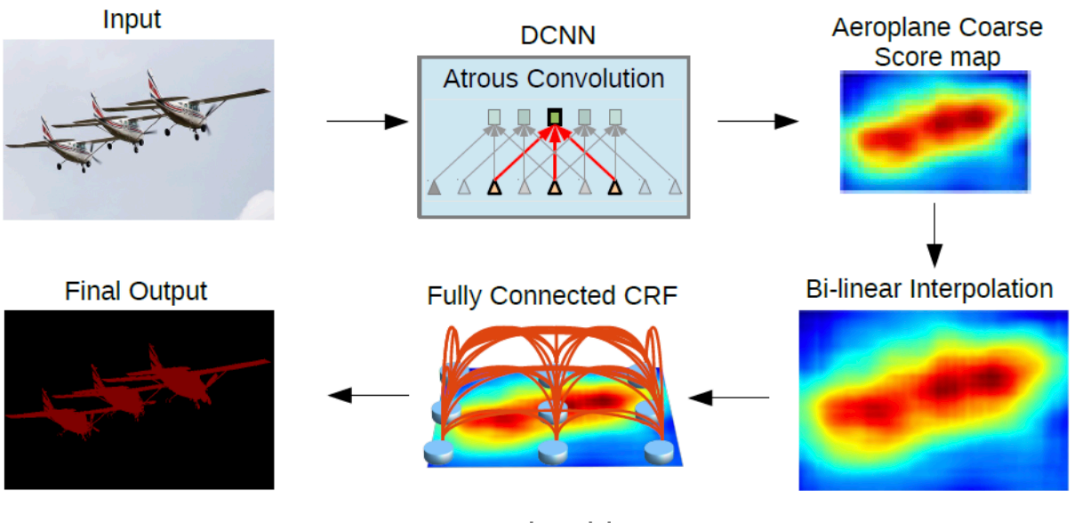
0.6.3. Deeplab

DeepLab --- это архитектура семантической сегментации. Сначала входное изображение проходит через сеть с использованием расширенных свёрток. Затем выходные данные сети билинейно интерполируются и проходят через полносвязный CRF для точной настройки результата, который мы получаем для окончательных прогнозов.

Основные характеристики:

1. **Ассоциативная CRF модель:** Одной из ключевых особенностей DeepLab является использование ассоциативной CRF (Conditional Random Field) модели для улучшения качества границ объектов. Эта модель обрабатывает карту сегментации и сглаживает границы между классами.
2. **Атрибутная декодирование:** DeepLab использует операцию декодирования для восстановления размера изображения и извлечения более детальной информации. Операция атрибутного декодирования позволяет извлекать более высокоуровневые признаки.
3. **Сверточные слои с большим шагом:** В архитектуре DeepLab применяются сверточные слои с большим шагом (dilated convolution), что позволяет увеличить приемное поле без потери пространственной разрешимости. Это помогает извлечь больше контекста.
4. **Использование предобученных моделей:** DeepLab может использовать предобученные модели, такие как ResNet, как основу для своей архитектуры. Это позволяет использовать предварительно обученные признаки и ускоряет процесс обучения.
5. **Multiscale аппроксимация:** Для учета объектов разных размеров DeepLab использует multiscale аппроксимацию, сочетая выходы разных слоев с разными масштабами. Это помогает достичь хорошей многомасштабной сегментации.

Архитектура DeepLab эффективно обеспечивает высокую точность сегментации объектов, а также устраняет артефакты и сглаживает границы. Её способность обрабатывать детали делает её подходящей для задач, требующих высокой степени точности.

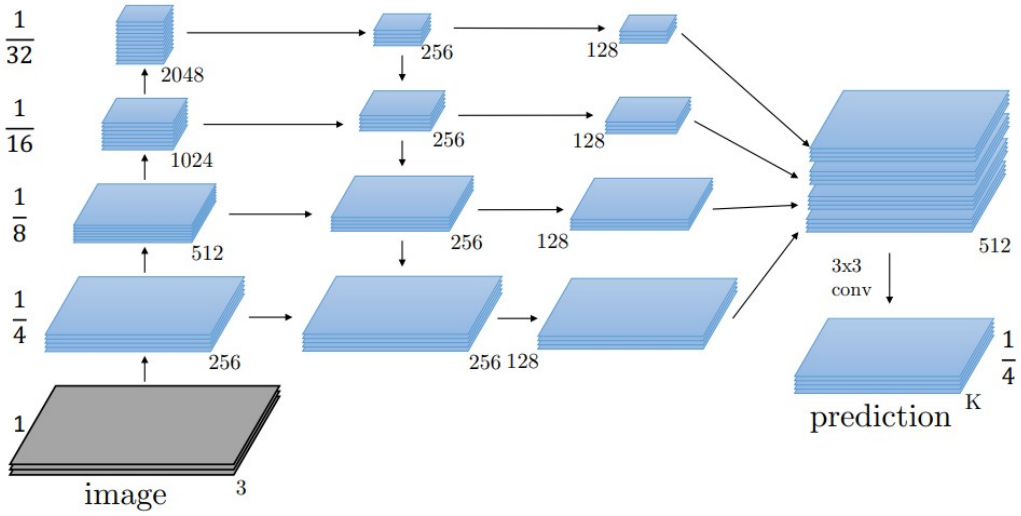


0.6.4. FPN (Feature Pyramid Network)

Архитектура FPN разработана для улучшения разномасштабной обработки изображений в сверточных нейронных сетях. Она позволяет моделям эффективно работать с объектами разных размеров, обеспечивая одновременно детализацию и контекст на разных уровнях масштаба.

Основные характеристики:

1. **Encoder**: Как и в U-Net архитектуре в начале идет путь свертки. В качестве энкодера может использоваться ResNet архитектура.
2. **Пирамида признаков:** Особенностью FPN является пирамида признаков, которая объединяет признаки с разных уровней свертки. Это позволяет модели улавливать контекст и детали разных размеров.
3. **Сквозные связи:** FPN также использует сквозные связи для передачи информации из энкодера в FCN блоки напрямую.
4. **Сегментация:** В результате объединения признаков формируются карты признаков разных масштабов



0.7. Экспериментальное исследование

Для каждой архитектуры CNN (U-Net, FCN, DeepLab, FPN) были выбраны соответствующие вариации моделей, учитывающие характерные особенности каждой архитектуры. Обучение проходило на обучающем наборе данных датасета ADE20K, который был разделен на обучающую и валидационную выборки.

В процессе обучения нейронных сетей была замечена особенность, связанная с функцией потерь DiceLoss. Данная функция потерь быстро достигала плато при обучении, что затрудняло дальнейшее улучшение результатов модели. В связи с этим, для повышения эффективности обучения и достижения лучших результатов, было принято решение использовать комбинацию функций потерь, CrossEntropyLoss и DiceLoss.

Для обучения были использованы оптимизатор Adam и механизм уменьшения скорости обучения ReduceLROnPlateau, контролирующий метрику Dice score.

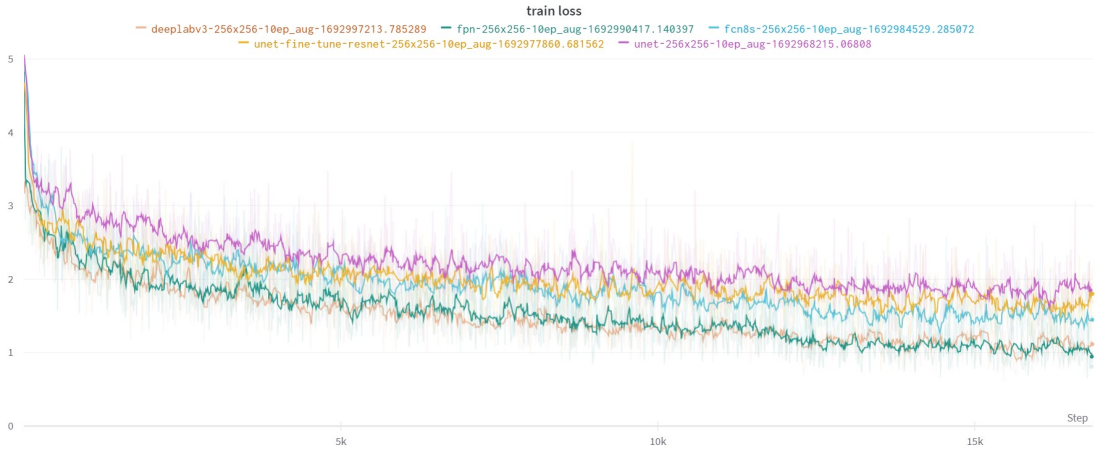
Для сохранения, анализа и визуального контроля результатов экспериментов была выбрана платформа Weights and Biases (wandb). Эта платформа предоставляет удобные инструменты для отслеживания процесса обучения, анализа метрик и визуальной оценки результатов.

Результаты всех экспериментов доступны по адресу: https://wandb.ai/losfantomaz/diploma

Из проведенного экспериментального исследования стало явно видно, что разные архитек-

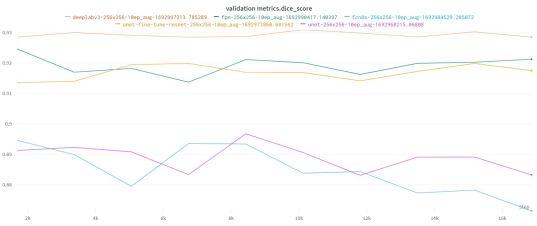
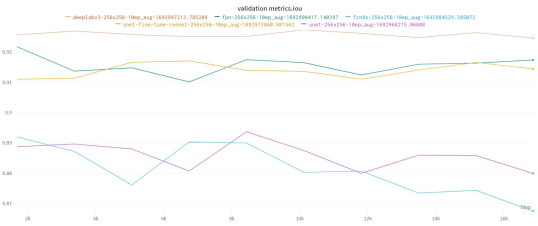
туры сверточных нейронных сетей (CNN) имеют разные скорости сходимости и качество сегментации в задаче обнаружения проходимых поверхностей.

Сравнение архитектур по скорости сходимости График функции потерь на этапе обучения позволяет сделать вывод, что архитектуры Deeplab и FPN сходятся быстрее к оптимальным значениям функции потерь по сравнению с U-Net и FCN.



Сравнение архитектур по метрикам Анализ показателей метрик подтвердил, что Deeplab демонстрирует лучшие результаты по всем метрикам: dice score, IoU и accuracy. Это говорит о том, что Deeplab обеспечивает наиболее точную и качественную сегментацию объектов на изображениях.

Архитектуры U-Net и FCN показали средние результаты по всем метрикам. В сравнении с Deeplab и FPN, эти архитектуры отстают как по качеству сегментации, так и по скорости сходимости. Это может быть связано с менее сложной архитектурой и недостаточным учетом деталей изображения.

Dice score	IoU
	

Результаты

Название модели	Размер изображения	mIoU	Dice	Accuracy	Precision	Recall	Train loss	Количество параметров
U-Net	256x256	0.8799	0.8832	0.9934	0.914	0.9555	2.52	31M

Название модели	Размер изображения	mIoU	Dice	Accuracy	Precision	Recall	Train loss	Количество параметров
U-Net fine-tuned	256x256	0.9144	0.9175	0.994	0.9507	0.956	3.749	24M
FCN	256x256	0.8675	0.8714	0.9945	0.8966	0.9595	1.382	18.7M
FPN	256x256	0.9174	0.9213	0.9951	0.9477	0.961	0.8848	23M
DeeplabV3	256x256	0.9246	0.9285	0.995	0.9557	0.961	1.269	26M

0.8. Выводы

Исследование показало, что выбор функции потерь является важным фактором при обучении нейронных сетей для семантической сегментации изображений. Подход с использованием комбинированной функции потерь, CrossEntropyLoss и DiceLoss, позволил улучшить качество сегментации. Однако, возможно, подбор другой функции потерь, также может привести к дополнительным улучшениям результатов. Например, функции потерь, ориентированные на борьбу с дисбалансом классов (Focal Loss, Lovasz Softmax Loss), могут быть рассмотрены для дальнейшего исследования.

Важным аспектом в разработке беспилотных систем и робототехнических приложений является комплексное использование различных методов и технологий. Нейронные сети могут быть успешно интегрированы с другими методами, такими как определение расстояний с помощью дальномеров и лидаров. Например, данные о расстояниях до препятствий, полученные с дальномеров, могут быть использованы в качестве дополнительных признаков для обучения нейронных сетей. Это позволит создать комплексную систему, способную более точно сегментировать поверхности.

Такой подход позволяет объединить преимущества разных методов и улучшить общую производительность системы. Комбинирование нейронных сетей с дополнительными датчиками и алгоритмами распознавания окружающей среды может обеспечить более надежное и точное решение задачи обнаружения препятствий и определения проходимых путей в реальных условиях.

0.9. Источники:

1. https://arxiv.org/abs/2209.06078
2. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. https://arxiv.org/pdf/1505.04597.pdf
3. Jonathan Long, Evan Shelhamer, Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. https://arxiv.org/pdf/1411.4038.pdf
4. Alexander Kirillov, Kaiming He, Ross Girshick, Piotr Dollár. A Unified Architecture for Instance and Semantic Segmentation. http://presentations.cocodataset.org/COCO17-Stuff-FAIR.pdf
5. https://arxiv.org/pdf/1706.05587.pdf
6. http://sceneparsing.csail.mit.edu/ ADE20K датасет