УДК 004.8

Салимли А.М. (4 курс бакалавриата, СПбПУ),
Востров А.В., старший преподаватель

ПОСТРОЕНИЕ ЗОНЫ ИНТЕРЕСА НА ОСНОВЕ СЕГМЕНТАЦИИ

Целью работы - исследование сегментаторов разных архитектур, выбор оптимального решения для построения зоны интереса (RoI). В рамках исследования проводятся подбор гиперпараметров, обучение и сравнение трёх моделей сегментации, а также разработка алгоритма построения RoI. В качестве зоны интереса рассматривается область между рельсами и прилегающие к ним участки по ширине трамвая.

Объектом исследования являются модели SegFormer, DeepLabV3+ с backbone ResNet101 и YOLO-seg. Данные модели основаны на различных архитектурных подходах: DeepLabV3+ использует глубокую нейронную сеть с использованием энкодера ResNet101, YOLO-seg основана на сверточной архитектуре (CNN) ориентированной на высокую скорость обработки, а SegFormer построен на трансформерной архитектуре. Сравнение моделей проводится с целью определения наиболее подходящего сегментатора для поставленной задачи по критериям точности не ниже 70% и скорости работы не менее 30 кадров в секунду (FPS).

Выборкой для обучения, был выбран открытый датасет из 8500 изображений с размеченными классами - RailSem19. Датасет содержит в себе сами изображения, разметку, в виде JSON формата, маски формата uint8. В датасете используется 19 классов, которые можно классифицировать на три составляющих множества: рельсы (5 классов), городская среда (11 классов), транспорт (4 класса)[2]. Обучение модели YOLO-seg, на таком классе невозможно, из-за несоответствия форматов данных, так как модель обучается на текстовых файлах формата $\{cls\ x_1 y_1\ x_2 y_2\ \dots\ x_n y_n\}$, где координаты описывают вершины полигона. Для решения данной проблемы, компания ultralytics, предлагает скрипт JSON2YOLO, который автоматически перевел датасет в нужный формат для обучения моделей YOLO-seg.

Для обучения сегментатора SegFormer, была выбрана конфигурация B0, которая содержит примерно 3,7 миллионов параметров [1]. Важно отметить, что SegFormer, как было сказано ранее работает на архитектуре трансформера с тяжелым энкодером и с легким MLP декодером. Энкодер работает по следующему принципу: к каждой точке сопоставляется вектор интенсивности RGB: $x(h, w) \in \mathbb{R}^3$, помимо цвета, точке может быть присвоено сгенерированные энкодером признаки которые в SegFormer, делятся на уровни по масштабу[2]. Далее применяется механизм внимания, который снижает вычислительную сложность за счет уменьшения размерностей ключей $Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_{head}}})V$, где Q - матрица запросов (информация о текущем элементе для которого мы ищем связи с другими элементами), K - матрица ключей которая содержит информацию об элементах последовательности с которыми сравниваются запросы (как метки), V - матрица значений которая содержит фактическую информацию которая будет суммироваться после вычисления весов внимания, d_{head} - длина векторов внимания, а для предотвращения слишком маленьких градиентов при обучении d_{head} вноситься под корень для масштабирования. Помимо этого SegFormer проделывает указанные операции по главной и второстепенной диагонали а так же по вертикали и горизонтали изображения, образуя карты признаков, и подаются на свертку

для получения тензора-внимания. Финальные веса применяются к исходным признакам и после финальный MLP-слой, предсказывает маску M размеренности $\frac{H}{4} \times \frac{W}{4} \times N_{cls}$[3].

Для начала обучения, были подобраны гиперпараметры с помощью метода tune[10] использующий эволюционную стратегию. Гиперпараметры представленны в таблице 1:

Таблица 1: Гиперпараметры для модели SegFormer B0

Гиперпараметр	Значение	Описание
model	SegFormer-B0	Заранее предобученная на датасете Ciryscapes
output_channels	14	Количество классов
batch_size	4	Размер мини-батчей
image_size	[1024,1024]	Размер изображения
optimizer	AdamW	Градиентный спуск с разделением затухания весов
loss_function	CrossEntropyLoss	Функция потерь кросс-энтропия
scheduler	LinearLR(start=1.0,final=0.5, total=30)	Шаги обучения
Epochs	150	Количество эпох обучения

На рисунке 1, показан спад функции потерь сегментации, как видно, на сотой эпохе, график вышел на плато, обучение дошло до 122 эпохи и завершилось досрочно.

На рисунке 2, изображен график величины шага корректировки весов в процессе оптимизации, видно, что на каждой итерации скорость (темп) обучения снижался вплоть до последней эпохи.

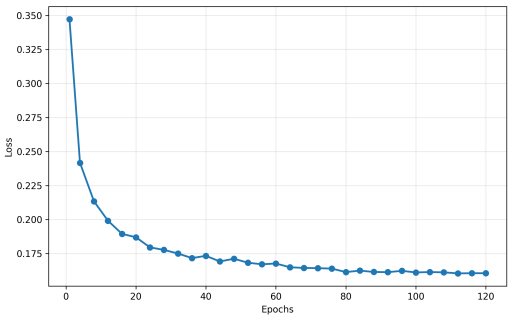

Рис.1: Значения функции потерь

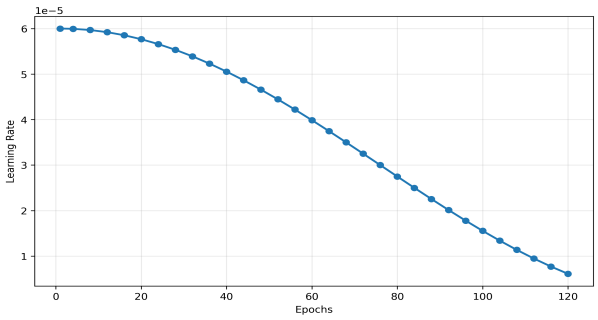

Рис.2: Скорость обучения

Так как конфигурация B0, является самой простой моделью сегментации ее точность по mIoU составила 0.651.

Следующая рассматриваемая модель DeepLabV3+ с энкодером ResNet101, главное отличие от обычной сверточной сети, заключается в атрозных (разряженных) сверток, для одномерного сигнала $x[i]$ атрозная свертка с фильтром $\omega[k]$ длиной K и скоростью расширения (dilation rate) r определяется как: $y[i] = \sum_{k=1}^{K} x[i + r \cdot k] \cdot \omega[k]$. В ResNet101

такой механизм, контролирует разрешение карт признаков. Модуль ASPP (Atrous Spatial Pyramid Pooling) позволяет захватывать контекст на разных масштабах.

При поступлении в декодер, признаки из ASPP апсэмплятся бинарной интерполяцией в 4 раза, далее уменьшаются количество признаков путем сверток для образования карт признаков, после все карты конкатинируются и применяется последний апсэмплинг который восстанавливает исходный размер изображени.

Функцией потерь в DeepLabV3+, является так же кросс-энтропия по пикселям: $L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} log(\hat{y_{i,c}})$, где N - число пикселей, C - число классов, y - истенная маска, $\hat{y}$ - предсказание модели. [4]

Модель DeepLabV3+, с энкодером ResNet101, по результатам оказалась худшей для сегментации рельс. Так как инференс модели составил более 500 мс., а точность по mIoU = 0.42.

В отличии от других архитектур, модели YOLO, имеет единый прогон по сети, который обеспечивает классификацию (cls), сегментацию (seg), выделение рамок (box), ориентировку (obb) и нахождение объекта (obj). Ссылаясь на источники, в качестве эксперемента будут обучены модели YOLOv8n-seg, YOLOv8s-seg, YOLOv11s-seg и YOLOv26s-seg[6][7][8][9].

Архитектура YOLO-seg представляется как композиция аппроксиматора признаков F, предсказывающего тензор масок (прототипов) $P \in \mathbb{R}^{h\times w\times k}$. Итоговая сегментация определяется линейной комбинацией $M = \sigma(P \cdot C^T)$, в качестве сигмоиды логистическая функция активации Оптимизация модели ведется через минимизацию функций потерь $\mathscr{L} = \lambda_1\mathscr{L}_{box} + \lambda_2\mathscr{L}_{cls} + \lambda_3\mathscr{L}_{seg}$. В поставленной задачи, важна функция потерь попиксельного соответствия предсказанных и истинных множеств $S \subset \Omega$, через бинарную кросс-энтропию. Модель YOLO-seg, работает на основе сверточных операций *CSPNet* для экстракции признаков. За объединение прихнаков отвечает блок *PANet*. Прототипы масок генерирует *Proto-Module*.

После получения гиперпараметров, было проведено обучение с стандартным различением датасета на три части: 7000 изображений на обучение, 1500 изображений на валидацию и 500 на тестирование моделей. Модель показавшая лучший результат, будет использована в дальнейшем для полного поиска гиперпараметров библиотекой Optune, и обучением с применением кросс-валидации.

По окончанию обучения и сравнения результатов на тестовой выборки, была выбрана модель YOLOv11s-seg. Точность модели по mIoU = 0.821, а скорость 32 FPS.

На рисунке 3 показаны результаты обучения моделей YOLOv11s-seg:

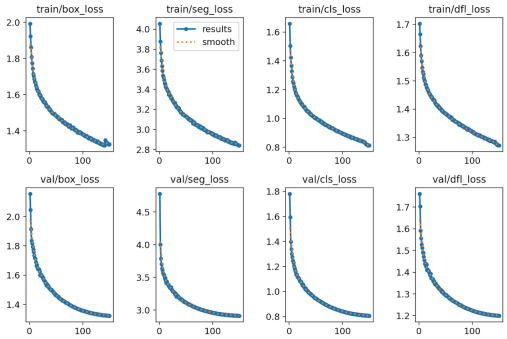

Рис.3: Результаты обучения

Для реализации алгоритма отображения зоны интереса (RoI), была использована следующая логика: на вход алгоритма поступает видеопоток, каждый кадр обрабатывается предобученой моделью YOLOv11s-seg. Из полученной карты классов выделяются пиксели, принадлежащие рельсовым объектам. Далее с помощью библиотеки OpenCV, применяется морфологическое закрытие и поиск связных компонент для выделения отдельных путей. Для каждой компоненты: определяется координаты левой и правой границ рельсов; границы сглаживаются скользящим среднем; с учетом масштаба строятся зоны опасности: красная (внутри рельсов +0.5м наружу), оранжевая (+1.0м), желтая (+1.5м) - путем расширения границ на соответствующее расстояние; зоны отображаются полупрозрачными полигонами и добавляется сетка с шагом 2 метра вдоль пути.

Данный алгоритм подойдет и для всех последующих моделей, так как на вход поступает сегментированные кадры, замена сегментатора не влияет на запуск алгоритма.

Результаты работы алгоритма представлены на рисунках 4-5.

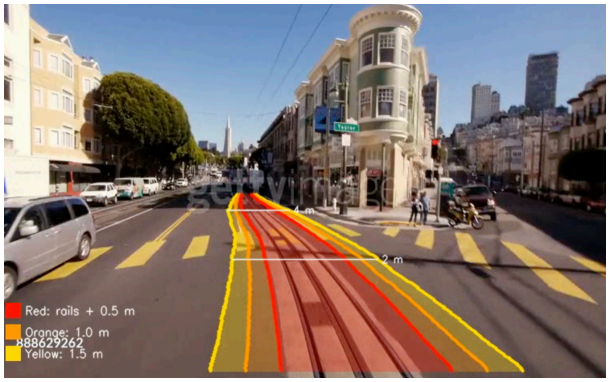

Рис.4: Пример работы алгоритма на кадре t_1

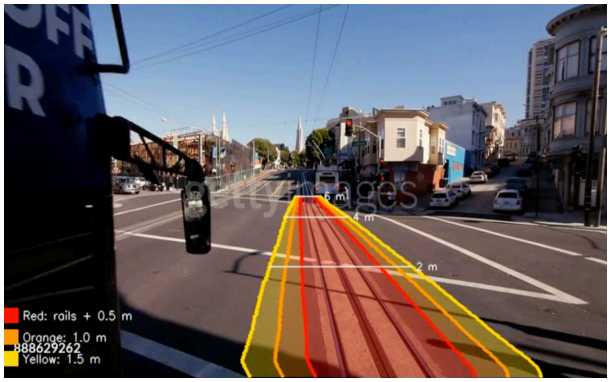

Рис.5: Пример работы алгоритма на кадре t_n

ЛИТЕРАТУРА

1.Xie, E. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers / E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, P. Luo // 35th Conference on Neural Information Processing Systems (NeurIPS 2021). – 2021.

2.Мороз, Е. С. Анализ SOTA-архитектур для семантической сегментации / Е. С. Мороз, Я. С. Пикалёв // Вестник Донецкого национального университета. Серия А: Естественные науки. - 2025. - № 4. - С. 39-60. - DOI 10.24412/2413-7383-2025-4-39-60-70.

3.arXiv:2105.15203 [cs.CV]. - URL: https://doi.org/10.48550/arXiv.2105.15203 (дата обращения: 21.03.2026). - Текст : электронный.

4.DeepLabV3Plus-RailSem19 [Электронный ресурс] : репозиторий / pideyi1025. – URL: https://github.com/pideyi1025/DeepLabV3Plus-RailSem19/tree/main (дата обращения: 21.03.2026).

5.A lightweight semantic segmentation model for autonomous driving based on Deeplabv3+ and attention mechanism / [authors] // Journal of Real-Time Image Processing. – 2023. – Vol. 20, no. 5. – DOI 10.1007/s11554-023-01380-x.

6.Ultralytics YOLO8 : сегментация [Электронный ресурс] // Ultralytics YOLO Docs. – URL: https://docs.ultralytics.com/ru/tasks/segment/#val (дата обращения: 21.03.2026).

7.Ultralytics YOLOv8 : ключевые особенности [Электронный ресурс] // Ultralytics YOLO Docs. – URL: https://docs.ultralytics.com/ru/models/yolov8/#key-features-of-yolov8 (дата обращения: 21.03.2026).

8.Ultralytics YOLOv11 : ключевые особенности [Электронный ресурс] // Ultralytics YOLO Docs. – URL: https://docs.ultralytics.com/ru/models/yolov11/#key-features-of-yolov11 (дата обращения: 21.03.2026).

9.Ultralytics YOLO26 : ключевые особенности [Электронный ресурс] // Ultralytics YOLO Docs. – URL: https://docs.ultralytics.com/ru/models/yolo26/ (дата обращения: 21.03.2026).

10.Ultralytics YOLO : гиперпараметры и настройка [Электронный ресурс] // Ultralytics YOLO Docs. – URL: https://docs.ultralytics.com/guides/hyperparameter-tuning/ (дата обращения: 21.03.2026).