УДК 004.852

А.В. Михеев, К.В. Святлов, Ю.А. Лапшов, В.Г. Тронин

ПРОГРАММНЫЕ СРЕДСТВА ДЛЯ АНАЛИЗА ПРИЗНАКОВ ТРАЕКТОРИИ ДВИЖЕНИЯ МОБИЛЬНОГО РОБОТА С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ С ВНИМАНИЕМ¹

Михеев Александр Вячеславович, окончил Ульяновский государственный технический университет. Программист Ульяновского областного центра новых информационных технологий при УлГТУ. Имеет более 15 статей, учебное пособие, свидетельство о регистрации программы для ЭВМ. Область научных интересов – машинное обучение. [e-mail: a.miheev@simcase.ru].

Святлов Кирилл Валерьевич, кандидат технических наук, окончил УлГТУ. Декан факультета информационных систем и технологий УлГТУ. Имеет 40 статей, 3 монографии, 3 учебных пособия, 4 свидетельства о регистрации программы для ЭВМ. Область научных интересов – машинное обучение и робототехника. [e-mail: k.svyatov@ulstu.ru].

Лапшов Юрий Александрович, кандидат технических наук, окончил УлГТУ. Доцент кафедры «Вычислительная техника» УлГТУ. Имеет 32 статьи, монографию, 2 свидетельства о регистрации программы для ЭВМ. Область научных интересов: системы автоматизированного проектирования, управление потоками проектных работ. [e-mail: y.lapshov@ulstu.ru].

Тронин Вадим Георгиевич, кандидат технических наук, окончил УлГТУ. Доцент кафедры «Информационные системы» УлГТУ. Имеет более 50 публикаций, учебное пособие. Область научных интересов: управление проектами, наукометрия, теория решения изобретательских задач. [e-mail: v.tronin@ulstu.ru].

Аннотация

В статье описан подход к реализации программного комплекса для определения вероятности столкновения мобильного робота с препятствиями, стоящими на пути его следования. Данный подход основан на нейросетевой модели с вниманием. Ключевой его особенностью является способ формирования обучающей выборки: разметка препятствий и значений вероятности столкновения с ними осуществляется не в ручном режиме, а с помощью детерминированного алгоритма, использующего результат семантической сегментации с применением другой предобученной нейронной сети. Данный способ позволяет использовать слабо детализованное описание внешней среды для обучения сверточных нейронных сетей с вниманием на примере распознавания преград при движении мобильного робота в режиме симуляции. При этом слабая детализация позволяет сократить трудоемкий процесс ручной разметки данных за счет автоматически генерируемой выборки в среде NVIDIA Isaac, а механизм внимания позволяет повысить интерпретируемость результатов анализа.

Ключевые слова: искусственный интеллект, нейронные сети, машинное обучение, компьютерное зрение, сети с вниманием, робототехника.

doi: 10.35752/1991-2927-2020-3-61-70-79

SOFTWARE FOR DETERMINING FEATURES OF A MOBILE ROBOT'S PATH WITH THE USE OF NEURAL NETWORKS

Aleksandr Viacheslavovich Mikheev, graduated from Ulyanovsk State Technical University; Programmer of the Ulyanovsk Regional Center of New Information Technologies at the Ulyanovsk State Technical University; an author of more than 15 research papers, a holder of the certificate of software registration; the area of his scientific interests relates to the machine learning. e-mail: a.miheev@simcase.ru.

Kirill Valerevich Sviatov, Candidate of Sciences in Engineering; graduated from Ulyanovsk State Technical University; Dean of the Faculty of Information Systems and Technologies of UISTU; an author of more than

¹ Исследование проводится при поддержке грантов РФФИ 18-47-732004 p_мк и 18-47-732012 p_мк.

40 articles, three monographs, three textbooks; a holder of four certificates of software registration; the area of his scientific interests relates to machine learning and robotics. e-mail: k.svyatov@ulstu.ru.

Iurii Aleksandrovich Lapshov, Candidate of Sciences in Engineering; graduated from Ulyanovsk State Technical University; Associate Professor of the Computer Engineering Department at UISTU; an author of 32 articles, a monograph; a holder of two certificates of software registration; the area of his scientific interests relates to computer-aided design systems, workflow management. e-mail: y.lapshov@ulstu.ru.

Vadim Georgievich Tronin, Candidate of Sciences in Engineering; graduated from Ulyanovsk State Technical University; Associate Professor of the Department of Information Systems at UISTU; an author of more than 50 articles, a textbook; the area of his scientific interests relates to the project management and theory of solving inventive problems. e-mail: v.tronin@ulstu.ru.

Abstract

The article describes an approach to the implementation of a software package for determining the probability of a collision of a mobile robot with obstacles. This approach is based on a neural network model with attention. Key feature is the method of the training dataset generation: the labeling of obstacles and the values of the probability of collision with them is performed not in manual mode, but using a deterministic algorithm that uses the result of semantic segmentation using another pre-trained neural network. This method allows to use a poorly detailed description of the external environment for training convolutional neural networks with attention on the example of recognizing obstacles when a mobile robot moves in simulation mode. At the same time, low detail allows to reduce the time-consuming process of manual data labeling due to automatically generated sampling in the NVIDIA Isaac environment, and the attention mechanism allows to increase the interpretability of the analysis results.

Key words: artificial intelligence, neural networks, machine learning, computer vision, attention networks, robotics.

ВВЕДЕНИЕ

В настоящее время важной особенностью задач, решаемых в процессе проектирования мобильных роботов, является необходимость использования эвристик и детерминированных алгоритмов в управляющих программах верхнего уровня логики в условиях большого количества параметров внешней для робота среды и недостаточным уровнем детализации таких параметров. Глубокие нейронные сети хорошо справляются с решением таких задач, но использование одной нейронной сети, решающей все задачи анализа сцен (так называемое «end-to-end» решение), невозможно из-за большой трудоемкости формирования обучающей выборки: каждый кадр видеопотока требует тщательной и сложной разметки. При этом архитектуры нейронных сетей, реализующих end-to-end подход, слишком сложны, а результат их работы хуже решений, использующих детерминированные алгоритмы. Также важным фактором при выборе архитектуры нейронных сетей в процессе реализации алгоритма являются высокие требования к производительности вычислительных устройств, особенно для мобильных роботов, в которых необходимо использовать оборудование с низким энергопотреблением.

Одним из первых этапов навигации мобильного робота является анализ внешней среды для построения ее цифровой модели. В ходе такого анализа определяется расположение препятствий на пути следования робота с использованием детерминированных алгоритмов, на вход которых подаются данные со специализированных дорогостоящих датчиков (например, лидаров). При мас-

совом производстве мобильной робототехники предпочтительнее использование обычных дешевых видеокамер. Кроме того, методы навигации, основанные на анализе видеоданных, позволяют использовать больше информации об окружении. За счет этого снижается интегральная ошибка при построении карты и локализации робота в пространстве. Но при анализе видеоданных отсутствует простой способ выделения детальных признаков для описания состояния робота, таких как динамические и статические препятствия на траектории робота, сама дорога или стены.

В данной работе рассматривается решение трех связанных между собой задач для выявления преград на пути следования мобильного робота:

1. Разработка программного средства для генерации специальной выборки данных. Предлагается способ генерации синтетических данных для обучения нейронной сети, способной распознавать преграды на изображении с камеры робота в среде NVIDIA Isaac.

2. Распознавание слабо детализированного описания признаков среды. Под слабо детализированным описанием признаков среды понимается бинарная характеристика, указывающая на опасность столкновения с препятствием с учетом текущей траектории движения робота или отсутствие такой опасности. При этом детализированное описание должно содержать полную информацию о препятствиях: расположение в кадре, размер, форма, класс объекта. То есть один и тот же объект может являться преградой, если он располагается на траектории следования робота, а может и не являться, если находится в отдалении. Поэтому, кроме факта наличия объекта, необходимо анализировать его

расположение относительно робота и другие характеристики. Критерии анализа могут быть сложными, учитывающими динамические свойства объектов, что влияет на время и стоимость разметки обучающей выборки в случае использования нейронных сетей и при разработке эвристических алгоритмов. В данной статье рассматриваются только статические объекты, подвижные объекты будут рассмотрены в следующих работах.

Далее описывается подход, в котором в качестве целевого признака обучающей выборки используется бинарный признак, свидетельствующий о том, есть ли препятствие на пути следования робота или нет: на нейронную сеть возлагается задача автоматического поиска закономерностей между сложными признаками объектов сцен с целью определения наличия препятствия. Это позволяет существенно снизить затраты на разметку сложных данных.

3. Обучение нейронной сети для локализации (определение положения) преград без учителя. В работе показывается, что сверточная нейронная сеть с «вниманием» способна научиться локализовывать преграды в кадре точнее, чем обычная сверточная сеть, а также позволяет локализовать объект в кадре без дополнительных затрат на разметку выборки по сравнению с задачей классификации. При этом полученную модель можно использовать для интерпретации результатов работы нейронной сети, т. е. для уточнения причин принятых ею решений, что крайне важно в задачах беспилотного управления мобильными роботами, когда принимаются решения, связанные с безопасностью людей или грузов.

СВЯЗАННЫЕ РАБОТЫ

Несмотря на успешность использования нейронных сетей в обработке изображений, они выполняют лишь определенные этапы такой обработки: обычно это детектирование объекта, его локализация на изображении, а также определение класса объекта. Также возможно определение всех имеющихся объектов на изображении и их последующий анализ, например, геометрическая оценка взаимного расположения объектов, их размера и формы. На следующем этапе принимается решение о параметрах управляющих воздействий для достижения роботом поставленных целей.

За последнее время было разработано большое количество моделей и подходов для детекции объектов на изображениях: это и сегментация с получением многоканальной маски на выходе [1, 2], и детекция с получением ограничивающих прямоугольников, и ряд модификаций этих методов по качеству и скорости [3–5].

Навигация робота осуществляется в динамике – состояния робота и среды постоянно меняются. Поэтому в разные моменты времени одни и те же объекты могут быть препятствиями, а могут и не представлять опасности для робота. При этом чем раньше будет предсказано, станет ли объект преградой, тем эффективнее можно составить алгоритм его объезда и даль-

нейшего планирования пути. Если рассматривать этот процесс в динамике, то подобный эффект схож с эффектом отложенных наград в обучении с подкреплением (Reinforcement learning), и в рамках данного типа обучения есть несколько методов для решения проблемы обучения модели нейронной сети при отложенных наградах или предсказания преграды заранее до непосредственного ее возникновения. В некоторых подходах нейронные сети выполняют весь цикл анализа сцен и принятия решений [6–9]. Но данное направление проработано меньше, чем распознавание объектов, из-за низкой интерпретируемости результатов работы сети: эксперту непонятно, на основании чего модель приняла те или иные решения. Кроме того, использование обучения с подкреплением само по себе является сложной задачей для реализации с рядом недостатков: нестабильность работы алгоритма, длительность обучения и других. Поэтому в данной работе основное внимание уделяется простым подходам, которые способны решить задачу быстрым в реализации «end-to-end» подходом с использованием нейронных сетей с вниманием. При этом описывается способ генерации обучающей выборки с использованием средств симуляции NVIDIA Isaac.

Механизм внимания в нейронных сетях [10] вызывает все больший интерес у исследователей [11], потому что позволяет интерпретировать результаты работы моделей и выполнять селекцию по произвольным позициям в картах признаков. Несмотря на то, что технология появилась несколько лет назад, спектр ее применения широкий [12].

ОПИСАНИЕ ПОДХОДА

В рамках рассматриваемого подхода детекцию преград при движении робота можно производить дискретно по времени: на каждой итерации цикла программного управления робот фиксирует состояние вокруг себя при помощи видеокамеры и совершает небольшое перемещение. При этом робот всегда движетсясонаправленно с камерой, то есть камера зафиксирована так, что ее фокус всегда совпадает с направлением движения робота в следующий момент времени. Поэтому объекты, расположенные по центру кадра и вблизи робота, могут представлять опасность и являются преградой (см. подробнее в разделе «Автоматическая разметка выборки для задачи классификации»).

Но эти критерии определения преград известны только для разработчика. Обучаемая модель нейронной сети не знает этих правил и допущений и получает на вход лишь бинарные метки 0 или 1, характеризующие наличие преграды. Задача нейронной сети заключается в предсказании наличия преграды с учетом ее положения, размера, типа объекта и всех возможных закономерностей, которая она сможет найти в данных самостоятельно.

Для решения поставленной задачи с использованием нейронной сети с вниманием необходима выборка

данных, которая представляет собой случайный кадр с камеры робота и бинарную метку наличия риска для следующих циклов программного управления роботом. Создание выборки данных является одной из основных проблем при проведении подобного рода исследований. Для задач классификации и сегментации [16] есть множество открытых наборов данных в сети, но для поставленной задачи оценки риска выборку данных найти не удалось. В рамках исследования было решено создать синтетическую выборку самостоятельно при помощи робототехнической платформы NVIDIA Isaac и встроенных в нее средств симуляции. За счет этого изображение с камеры робота было заменено на рендер виртуального 3D-мира с фотореалистичным качеством. Ранее была проделана работа, которая показывает, что современный уровень графики игр способен служить заменой реальному миру для создания обучающей выборки [13].

Но стандартные инструменты не способны определить вероятность риска движения для робота, поэтому был реализован программный компонент для автоматической разметки выборки на основании сегментационных масок сцены, полученных также из NVIDIA Isaac.

Далее было произведено обучение двух нейронных сетей: обычной сверточной сети архитектурой VGG и сверточной сети с визуальным вниманием без рекуррентного блока в стиле [14].

Сверточная нейронная сеть аппроксимирует распределение $P(y|x)$, где целевая переменная y – это бинарное значение, определяющее наличие или отсутствие преграды, а x – это входная матрица пикселей изображения. Для известных пар y и x , нейронная сеть максимизирует функцию правдоподобия путем подстройки параметров θ :

$$\max_{\theta} L(\theta|y, x). \quad (1)$$

Механизм выравнивания, который позднее стал общим механизмом внимания, был предложен в работе Yoshua Bengio и других авторов [11]. В этой статье было предложено улучшить автоматический перевод текстов с помощью добавления веса для скрытого состояния энкодера в каждый момент времени. До внедрения этого механизма декодер итеративно генерировал слова посредством аппроксимации рекуррентной функции вида:

$$P(y_t|\{y_1, \dots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c), \quad (2)$$

где y_t – это слово из некоторого словаря D_y в момент t ; g – это нелинейная функция, например, многослойная нейронная сеть, которая выводит вероятность слова y_t на основании текущего состояния s_t декодера RNN (recurrent neural network, рекуррентная нейронная сеть) и c , c – это вектор, генерируемый из последовательности скрытых состояний энкодера как интегральная величина:

$$\begin{aligned} h_t &= f(x_t, h_t - 1), \\ c &= q(\{h_1, \dots, h_{T_x}\}), \end{aligned} \quad (3)$$

где $h_t \in R$ – это скрытое состояние энкодера RNN сети в момент времени t ; f и q – некоторые нелинейные функции. Например, Sutskever et al. (2014) использовали модель LSTM (Long Short Term Memory, модель с долгой кратковременной памятью) в качестве f и q . Авторы механизма внимания предложили учитывать контекст энкодера в каждый момент времени при работе декодера (вместо интегрального c используется c_i):

$$P(y_i|\{y_1, \dots, y_{i-1}\}, x) = g(y_{i-1}, s_i, c_i) \quad (4)$$

на основании взвешенной суммы:

$$c_i = \sum_{j=1}^{T_x} a_{ij} h_j, \quad (5)$$

где a и есть вес текущего состояния энкодера (т. е. важность конкретного j -го элемента h для i -го элемента контекста c). Нормированный вес можно рассчитать функцией softmax:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \quad (6)$$

$$e_{ij} = \text{att}(s_{i-1}, h_j).$$

Эта модель оценивает, насколько сильно связаны входные элементы вокруг позиции j и выходные элементы вокруг позиции i . Эта оценка основана на скрытом состоянии RNN s_{i-1}, j -й и на основе входной последовательности h при помощи еще одной нелинейной функции att , которая представляет собой полносвязную нейронную сеть. При этом параметры полносвязной нейронной сети, которая совместно обучается со всеми остальными нейронами сверточной сети.

Таким образом, происходит уточнение вероятностных оценок на выходе сети за счет учета скрытых состояний в каждый момент времени.

В предложенной нами сети нет множества состояний (т. е. она не рекуррентная), однако соотношения 5–6 можно использовать для получения карт признаков в разных позициях. Это позволит лучше аппроксимировать целевую функцию, а также построить карту α , которая будет принимать высокие значения в тех локациях (пикселях), где признаки сильнее коррелируют с выходом сети. Эту карту можно интерпретировать как «внимание» нейронной сети при выводе.

ГЕНЕРАЦИЯ БАЗОВОЙ ВЫБОРКИ

Для генерации обучающей выборки было решено использовать программный фреймворк NVIDIA Isaac. Набор инструментов для разработчиков (SDK) NVIDIA Isaac содержит программные инструменты, библиоте-

ки, алгоритмы для GPU и компоненты, позволяющие ускорить разработку программных компонентов для роботов. Nvidia ISAAC также может быть использован для разработки готовых к производству решений с искусственным интеллектом, оптимизированных для развертывания на платформе NVIDIA Jetson.

Схожим с NVidia ISAAC функционалом обладают другие робототехнические фреймворки, такие как V-Rep, ROS (со средой симуляции Gazebo), но преимуществом ISAAC SDK по сравнению с ними является использование фреймворка Unity 3D для фотореалистичной симуляции среды и роботов (ISAAC SIM), а также богатый набор библиотек для решения задач автономного управления, наличия предобученных нейронных сетей, а также программных возможностей Cuda SDK для расчетов параметров движения и анализа сцен. За счет этого может быть реализовано существенное ускорение расчетов.

Задача генерации обучающей выборки была сформулирована следующим образом: необходимо разработать программный компонент для генерации в среде симуляции множества пар изображений, каждая пара в котором представляет собой два изображения: кадр снимка с цветной камеры и изображение с цветовой семантической сегментацией, при которой каждый пиксель сегментационной маски является отображением числа от 1 до 5 в значение кортежа RGB (красный, зеленый, синий цвета), соответствующего одному из пяти классов:

- Свободное для перемещения пространство (floor),
- Линия дорожной разметки (laneline),
- Препятствие (obstacle),
- Стена (wall),
- Балка (beam).

Для реализации этой задачи в среде Isaac Sim использовалась сцена «Medium Warehouse» и стандартный скрипт, который позволяет случайным образом изменять положение камеры, осуществлять «телепортацию» (изменение положения, наклона, угла поворота камеры). Кроме того, использовалась сегментационная нейронная сеть с архитектурой UNet (стандартный компонент Isaac SDK).

До запуска команд «телепортации» необходимо из директории симулятора запустить саму среду Unity 3D, затем запустить процесс дообучения сегментационной нейронной сети.

После выполнения команд запуска открывается окно с визуализацией сцены хранилища (рис. 1).

При запуске приложения (Codelet) в среде Isaac SDK запускается сервер WebSight (стандартное приложение визуализации) на порту 3000, который позволяет отображать графики, 2D- и 3D-сцены, текущее состояние приложения (активные узлы и соединения), а также обновлять конфигурацию. При запуске приложения «freespacednn» (поиск свободной дороги для перемещения робота) в среде WebSight выводятся изображения с камеры (последовательные снимки камеры

Рис. 1. Визуализация сцены хранилища (Medium Warehouse)

в процессе «телепортации»), а также результат работы сегментационной сети с архитектурой UNet (файл с определением модели сети – `packages/freespace_dnn/apps/freespace_dnn_training_models.py`)

Изображения с камеры и результат работы сегментационной сети отображаются в среде WebSight по каналам `freespace_dnn_training_unity3d/training_color_camera_viewer/ColorCameraViewer/Color` и `freespace_dnn_training_unity3d/segmentation_viewer/Segmentation/`.

Целевая обучающая выборка должна хранить в папках `color` и `seg` файлы с одинаковыми именами для соответствующих друг другу изображений и карты сегментации. Для того, чтобы сохранить изображение с камеры и результат работы сегментационной сети в файлы, необходимо было разработать приложение в среде Isaac SDK, использующее возможности этой платформы.

Разработанное приложение состоит из четырех файлов:

- `__init__.py` – служебный файл, необходимый для запуска программного пакета, он может быть пустым;
- `BUILD` – файл сборки для программной системы компиляции и запуска приложений Bazel;
- `seg_dataset.app.json` – конфигурационный файл приложения, оформленный в соответствии с требованиями ISAAC SDK;
- `seg-dataset.py` – программный код приложения.

В файле `BUILD` прописываются файлы, из которых состоит разрабатываемый программный компонент, а также параметры для его запуска и исполнения.

В скрипте `seg_dataset.py` реализуется сбор сообщений с камеры робота, сообщений от сегментационной сети, а также формирование изображений сегментации, и сохранение всех изображений в целевую директорию с учетом правил именования (одинаковые имена файлов для соответствующих изображений в папках `seg` и `image`). Для получения изображений от камеры и от сегментационной сети необходимо создать объекты, соответствующие каналам для цветного изображения и данным сегментации соответственно:

```
self.rgb_rx = self.isaac_proto_rx(«ColorCameraProto», «color»)
self.seg_rx = self.isaac_proto_rx(«SegmentationCameraProto», «segmentation»)
```

Точкой входа в программный цикл управления является метод `tick`, который вызывается автоматически объектом типа `Application` при регистрации разработанного приложения в среде Isaac SDK. Проверка наличия изображения в канале данных осуществляется методом `self.rgb_rx.available()`.

```
rgb_image = np.frombuffer(rgb_image_buffer, dtype=rgb_data_type)
rgb_image = rgb_image.reshape((rgb_rows, rgb_cols, rgb_channels))
im = Image.fromarray(rgb_image)
im = im.resize((420, 240), Image.ANTIALIAS)
im.save(«/images/color/robot_image_{}.png».format(num))
```

Аналогично осуществляется считывание сегментационной маски с той лишь разницей, что для формирования цветного изображения сегментации (3 канала RGB для каждого пиксела) из матрицы, каждый элемент которой помечен одним скалярным значением, соответствующим порядковому номеру класса, необходимо осуществить простое преобразование этого изображения:

```
width, height, channels = image.shape
out = np.empty((width, height, 3), dtype=np.uint8)
image = np.stack(image, axis=-1)
image = np.stack(image, axis=-1)
# 1 = красный      255    0    0
# 2 = синий        0     0    255
# 3 = зеленый      0     255  0
# 4 = голубой     0     255  255
# 5 = желтый      255    255  0
red = lambda x: 255 if (x == 1) or (x == 5) else 0
green = lambda x: 255 if (x == 3) or (x == 4) or (x == 5) else 0
blue = lambda x: 255 if (x == 2) or (x == 4) else 0

out[:, :, 0] = np.vectorize(red)(image)
out[:, :, 1] = np.vectorize(green)(image)
out[:, :, 2] = np.vectorize(blue)(image)
```

В итоге был получен набор синтетических данных, где x – рендер сцены, которая имитирует данные видеокамеры, расположенной на борту мобильного робота, а y – соответствующие маски объектов. Каждый класс объектов сцены представлен на маске своим цветом, что является результатом автоматической сегментации (рис. 2).

АВТОМАТИЧЕСКАЯ РАЗМЕТКА ВЫБОРКИ ДЛЯ ЗАДАЧИ КЛАССИФИКАЦИИ

После получения базовой выборки необходимо изменить целевую переменную y таким образом, чтобы она представляла собой не маску сцены, а бинарную

метку (0 или 1), указывающую на наличие опасного для движения робота препятствия в следующие циклы программного управления.

Если изображение доступно для считывания, то само считывание осуществляется методом `rgb_image_proto = self.rgb_rx.get_proto().image` в переменную `rgb_image_proto`.

Сохранение осуществляется стандартными средствами языка python:

Прежде чем решать задачу в динамике для произвольного поведения камеры, были сделаны допущения о том, что текущая траектория движения робота совпадает с центром изображения с камеры робота. Такое допущение является типовым для задач беспилотного управления в мобильной робототехнике.

Для бинаризации целевого параметра была предложена простая гипотеза: если объекты, относящиеся к классу преграды, находятся в центральной зоне кадра и занимают достаточную большую площадь, то они являются преградой, если они находятся в другом месте или занимают маленькую площадь, то они не являются преградой. При наличии сегментационной маски данную гипотезу оказалось легко реализовать при помощи библиотеки OpenCV.

Для бинаризации целевого параметра была предложена простая гипотеза: если объекты, относящиеся к классу преграды, находятся в центральной зоне кадра и занимают достаточную большую площадь, то они являются преградой, если они находятся в другом месте или занимают маленькую площадь, то они не являются преградой. При наличии сегментационной маски данную гипотезу оказалось легко реализовать при помощи библиотеки OpenCV.

Рис. 2. В верхней части рисунка (а, б) представлены примеры рендера виртуального мира. В нижней части рисунка (в, г) – соответствующие этому рендеру маски объектов. Каждому типу (классу) объектов на сцене соответствует свой цвет

Алгоритм можно представить следующим образом на псевдокодовом языке:

```
function calc_risk(segmentation_mask, triangle_points, area_threshold):
    // получение треугольной маски области интересов
    triangle_mask = draw_triangle_mask(triangle_points)
    // получение области интересов путем наложения маски и сегментации
    roi = image_bitwise_and(segmentation_mask, triangle_mask)
    // фильтрация области интересов по цветам для определения препятствий
    _, blocks_clr_mask = filter_non_blocks_colors(roi)
    // определение площади изображения, занимаемой препятствиями
    risk_area = count_nonzero(blocks_clr_mask)
    risk_status = 0
    // возврат значения пороговой функции
    if risk_area > area_threshold:
        risk_status = 1
    return risk_status
```

В качестве промежуточного этапа исполнения этого алгоритма будут получаться следующие сегментационные маски (см. рис. 3), на основе которых в заданной зоне интереса вычисляется площадь рассматриваемого объекта и оценивается итоговый риск. Простая пороговая функция определяет, является ли объект преградой или нет.

Рис. 3. Внешний вид зоны интереса (переменной roi в псевдокоде) для оценки риска движения. По данной зоне на кадре оценивается площадь потенциальных преград и проставляется бинарная метка

ОПИСАНИЕ ЭКСПЕРИМЕНТОВ И РЕЗУЛЬТАТЫ

На входе обеих моделей нейронных сетей использовалось только визуальное представление сцены без дополнительной информации об объектах. На выходе сети ожидался флаг, характеризующий риск для движения робота.

Каждая модель была запущена 10 раз для статистической уверенности в результатах. Обучение производилось на 15 эпохах, после которых происходило насыщение моделей, batch_size 32, всего выборка состояла из 3511 изображений, 33% из которых были валидационными, остальные – тренировочные. После каждой эпохи тренировочная выборка перемешивалась (shuffle = True), а между каждым запуском (из 15 итераций) выборка заново случайным образом разбивалась на тренировочную и валидационную.

Из таблицы видно, что даже обычная сверточная нейронная сеть успешно нашла корреляцию между

объектами, их положением в кадре и риском. А сверточная сеть, дополненная механизмом внимания, работает не только точнее, но и позволяет еще локализовать эти объекты без дополнительной разметки обучающей выборки, т. е. косвенно обучается на задаче классификации.

В таблице указана точность распознавания наличия препятствия на обучающей выборке (Train Accuracy) и на валидационной выборке (Valid Accuracy), где μ – среднее значение, σ – стандартное отклонение. В таблице приведено сравнение для двух типов нейронных сетей: нейронная сеть с вниманием (ATT) и обычная сверточная нейронная сеть (CNN).

Таблица

Точность распознавания препятствия при движении

Метрика	CNN	ATT
Train Accuracy, %	$\mu = 0,949$ $\sigma = 0,142$	$\mu = 0,981$ $\sigma = 0,005$
Valid Accuracy, %	$\mu = 0,939$ $\sigma = 0,145$	$\mu = 0,947$ $\sigma = 0,015$

Сеть с вниманием позволяет получить на выходе не только метку класса, но и карту внимания. Эта карта вещественных значений, где высокие значения означают большее влияние соответствующих пикселей на выход сети. Уже после нескольких эпох обучения визуализация карт внимания показывает, что высокие значения сосредотачиваются на нужных объектах. На рисунке 4 видно, что когда объект расположен по ходу движения робота, то он классифицируется как преграда (б, г), а когда объект расположен далеко от центра движения, то игнорируется (а, в). При этом внимание покрывает потенциальные преграды независимо от их расположения, но бинарная оценка при этом определяется верно. На нескольких кадрах потенциальных преград практически не видно. Именно такое поведение и ожидалось от системы.

Рис. 4. Внешний вид тепловых карт внимания, наложенных поверх исходных изображений сцены. Белыми пятнами внимание выделяет потенциальные преграды в кадре. Преграды помечены черным цветом. Возможна и обратная конфигурация, когда внимание будет выделять преграды

Принцип вычислений механизма внимания представлен на рисунке 5 [15]. На вход нейронной сети подаются карты признаков от выхода сверточной сети, которые преобразуются из тензора в вектор, в результате чего получается набор векторов длиной T в количестве L . Затем вектор из элементов каждой карты в позиции t матрично умножается на вектор внимания a , размерности L , затем генерируется вектор z размерности T . После нормализации функцией softmax генерируется распределение значимости областей на исходном изображении. Полученный вектор вероятностей затем умножается на исходные карты признаков, нормализуя их по важности в каждой позиции. Чем выше значение элемента в маске внимания, тем выше корреляция признаков в данной области изображения с выходом нейронной сети. Обычная операция свертки определяет корреляцию признаков с некоторым банком фильтров по разным позициям входного изображения. Операция внимания, во-первых, объединяет все признаки в одну интегральную оценку, а во-вторых, производит масштабирование этих признаков по каждой позиции

Рис. 5. Механизм внимания [15]

функцией softmax. В процессе обучения это вынуждает сеть отдавать высокие оценки тем позициям, в которых присутствуют наиболее важные признаки. В случае, когда модель решает задачу классификации, с каждой выходной меткой (классом) будет коррелировать определенная маска, которую и можно использовать в качестве сегментационной маски объекта. В тех позициях, где признаки оказались малозначимыми для выходной метки, маска внимания принимает значение ноль или очень близкое к нулю.

Далее осуществлялись бинаризация маски с использованием пороговой функции и выделение контуров предполагаемых объектов (преград). Результат такого выделения представлен на рисунке 6. При достаточной точности модели можно локализовать объекты в кадре и оценивать их параметры (площадь, конкретный класс объекта и т. п.), тем самым повышая степень детализации распознавания сцен за счет выявления иерархии признаков. Весь процесс был реализован без ручной разметки масок сегментации для обучения сети.

Рис. 6. Выделение объектов с использованием внимания. Объектом считалась связанная область со значением внимания выше 10% от максимального уровня маски внимания. На кадре внизу слева видно, что шкафы выделились достаточно точно. Ящик на кадре внизу справа выделился фрагментарно

В данной работе не производилось измерение качества выделения контуров объектов (преград). Сам подход оказался работоспособным, при этом и точность полученного результата может быть улучшена.

Выводы

В статье описан простой в использовании подход для генерации синтетической выборки с автоматической разметкой с использованием робототехнической платформы NVIDIA Isaac для задач анализа сцен при движении мобильных роботов. В предлагаемом подходе задача детекции преград решалась как задача бинарной классификации с использованием модели сверточной нейронной сети «end-to-end» с точностью 93,9%

без сложной разметки данных или разработки дополнительных эвристик.

При помощи механизма внимания разработанный подход может быть использован для получения дополнительных признаков преград: расположение объектов в кадре, их размеры. Использование внимания повышает точность распознавания преград в слабо детализированной постановке.

Дальнейшая работа может быть направлена на повышение точности локализации и определения пространственных свойств преград (размера, контуров, дистанции до объекта и т. п.), а также прогнозирования ситуаций столкновения динамических объектов.

При этом слабую детализацию целевой переменной можно применять в широком спектре задач распознавания и анализа сцен в дополнение к распознаванию преград для мобильных роботов. А механизм внимания позволяет выделять требуемые объекты, влияющие на результат работы сети с использованием слабо детализированной выборки данных.

СПИСОК ЛИТЕРАТУРЫ

1. Xiaolong Liu, Zhidong Deng, Yuhan Yang. Recent progress in semantic image segmentation. URL: <https://arxiv.org/ftp/arxiv/papers/1809/1809.10198.pdf> (дата обращения: 01.07.2020).
2. Rich feature hierarchies for accurate object detection and semantic segmentation / R. Girshick, J. Donahue, T. Darrell, J. Malik. URL: <https://arxiv.org/abs/1311.2524> (дата обращения: 01.07.2020).
3. Tech report (v5). URL: <https://arxiv.org/pdf/1311.2524.pdf> (дата обращения: 13.07.2020).
4. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon, S. Divvala, R. Girshick, A. Farhadi. URL: <https://arxiv.org/pdf/1506.02640.pdf> (дата обращения: 13.07.2020).
5. Girshick R. Fast R-CNN. URL: <https://arxiv.org/pdf/1504.08083.pdf> (дата обращения: 13.07.2020).
6. Object Detection with Deep Learning: A Review / Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, Xindong Wu. URL: <https://arxiv.org/pdf/1807.05511.pdf> (дата обращения: 15.07.2020).
7. Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates / S. Gu, E. Holly, T. Lillicrap, S. Levine. URL: <https://arxiv.org/pdf/1610.00633.pdf> (дата обращения: 13.07.2020).
8. End-to-End Robotic Reinforcement Learning without Reward Engineering / A. Singh, L. Yang, K. Hartikainen, C. Finn, S. Levine. URL: <https://arxiv.org/pdf/1904.07854.pdf> (дата обращения: 10.07.2020).
9. FeUdal Networks for Hierarchical Reinforcement Learning / A.S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, K. Kavukcuoglu. URL: <https://arxiv.org/pdf/1703.01161.pdf> (дата обращения: 07.07.2020).
10. Srinivas A., Laskin M., Abbeel P. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. URL: <https://arxiv.org/pdf/2004.04136.pdf> (дата обращения: 08.07.2020).

11. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention / K. Kelvin Xu, J.L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R.S. Zemel, Y. Bengio. URL: <https://arxiv.org/pdf/1502.03044.pdf> (дата обращения: 08.07.2020).

12. Learning Unsupervised Video Object Segmentation through Visual Attention / W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S.C.H. Hoi, H. Ling. URL: http://openaccess.thecvf.com/content_CVPR_2019/papers/Wang_Learning_Unsupervised_Video_Object_Segmentation_Through_Visual_Attention_CVPR_2019_paper.pdf (дата обращения: 13.07.2020).

13. Effective Use of Synthetic Data for Urban Scene Semantic Segmentation / F.S. Saleh, M. Sadegh, M. Salzmann, L. Petersson, J.M. Alvarez. URL: <https://arxiv.org/pdf/1807.06132.pdf> (дата обращения: 02.07.2020).

14. SAUNet: Shape Attentive U-Net for Interpretable Medical Image Segmentation / J. Sun, F. Darbehani, M. Zaidi, Bo Wang. URL: <https://arxiv.org/pdf/2001.07645v3.pdf> (дата обращения: 01.08.2020).

15. Scenes Segmentation in Self-driving Car Navigation System Using Neural Network Models with Attention / K. Sviatov, A. Miheev, D. Kanin, S. Sukhov, V. Tronin ; S. Misra et al. (eds) // Computational Science and Its Applications – ICCSA 2019. Lecture Notes in Computer Science. 2019. Vol 11623. Springer, Cham.

16. Comparative Study of Real-time Semantic Segmentation for Autonomous Driving / M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, H. A. Zhang. URL: http://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w12/Siam_A_Comparative_Study_CVPR_2018_paper.pdf (дата обращения: 13.07.2020).

REFERENCES

1. Xiaolong Liu, Zhidong Deng, Yuhan Yang. *Recent Progress in Semantic Image Segmentation*. Available at: <https://arxiv.org/ftp/arxiv/papers/1809/1809.10198.pdf> (accessed 01.07.2020).
2. Girshick R., Donahue J., Darrell T., Malik J. *Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation*. Available at: <https://arxiv.org/abs/1311.2524> (accessed 01.07.2020).
3. *Tech Report (v5)*. Available at: <https://arxiv.org/pdf/1311.2524.pdf> (accessed 13.07.2020).
4. Redmon J., Divvala S., Girshick R., Farhadi A. *You Only Look Once: Unified, Real-Time Object Detection*. Available at: <https://arxiv.org/pdf/1506.02640.pdf> (accessed 13.07.2020).
5. *Girshick R. Fast R-CNN*. Available at: <https://arxiv.org/pdf/1504.08083.pdf> (accessed 13.07.2020).
6. Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, Xindong Wu. *Object Detection with Deep Learning: A Review*. Available at: <https://arxiv.org/pdf/1807.05511.pdf> (accessed 15.07.2020).
7. Gu S., Holly E., Lillicrap T., Levine S. *Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates*. Available at: <https://arxiv.org/pdf/1610.00633.pdf> (accessed 13.07.2020).

8. Singh A., Yang L., Hartikainen K., Finn C., Levine S. *End-to-End Robotic Reinforcement Learning without Reward Engineering*. Available at: <https://arxiv.org/pdf/1904.07854.pdf> (accessed 10.07.2020).
9. Vezhnevets A.S., Osindero S., Schaul T., Heess N., Jaderberg M., Silver D., K. Kavukcuoglu. *FeUdal Networks for Hierarchical Reinforcement Learning*. Available at: <https://arxiv.org/pdf/1703.01161.pdf> (accessed 07.07.2020).
10. Srinivas A., Laskin M., Abbeel P. *CURL: Contrastive Unsupervised Representations for Reinforcement Learning*. Available at: <https://arxiv.org/pdf/2004.04136.pdf> (accessed 08.07.2020).
11. K. Kelvin Xu, J.L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R.S. Zemel, Y. Bengio. *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. Available at: <https://arxiv.org/pdf/1502.03044.pdf> (accessed 08.07.2020).
12. W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S.C.H. Hoi, H. Ling. *Learning Unsupervised Video Object Segmentation through Visual Attention*. Available at: http://openaccess.thecvf.com/content_CVPR_2019/papers/Wang_Learning_Unsupervised_Video_Object_Segmentation_Through_Visual_Attention_CVPR_2019_paper.pdf (accessed 13.07.2020).
13. Saleh F.S., M. Sadegh, M. Salzmann, L. Petersson, J.M. Alvarez. *Effective Use of Synthetic Data for Urban Scene Semantic Segmentation*. Available at: <https://arxiv.org/pdf/1807.06132.pdf> (accessed 02.07.2020).
14. Sun J., F. Darbehani, M. Zaidi, Bo Wang. *SAUNet: Shape Attentive U-Net for Interpretable Medical Image Segmentation*. Available at: <https://arxiv.org/pdf/2001.07645v3.pdf> (accessed 01.08.2020).
15. Sviatov K., Mikheev A., Kanin D., Sukhov S., Tronin V., Misra S. et al. (eds). *Scenes Segmentation in Self-driving Car Navigation System Using Neural Network Models with Attention. Computational Science and its Applications – ICCSA 2019. Lecture Notes in Computer Science*. 2019, vol. 11623, Springer, Cham.
16. Siam M., M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, H. A. Zhang. *Comparative Study of Real-time Semantic Segmentation for Autonomous Driving*. Available at: http://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w12/Siam_A_Comparative_Study_CVPR_2018_paper.pdf (accessed 13.07.2020).