ISSN 2079-3316 **ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ** **т. 9**, №4(39), с. 279–291

ББК 32.972.13
УДК 004.932.72'1

А. В. Смирнов, Д. Н. Степанов

Распознавание типа внутренних элементов планировки зданий в задачах первичной навигации мобильных роботов с использованием нейронных сетей

Аннотация. В статье предложен метод распознавания свёрточной нейронной сетью (СНС) типа внутренних элементов планировки здания с использованием особых черт изображений этих элементов, таких как границы или края. Предлагается алгоритм предобработки изображений для выделения границ на изображении. Создаётся база данных с изображениями различных типов внутренних элементов планировки здания, таких как коридор, дверь (дверной проём), угловые конструкции и лестницы. Также в статье представлена собственная структура СНС для распознавания типов помещений, и представлены данные о точности распознавания. Разработанный метод предлагается использовать для первичной навигации мобильных роботов.

Ключевые слова и фразы: свёрточная нейронная сеть, распознавание помещений, фильтрация изображений, блочно-параллельная обработка, выделение особых черт.

Введение

В наше время в сфере информационных технологий всё чаще применяется нейросетевой подход для решения многих задач распознавания и прогнозирования. Широкое распространение в задачах распознавания получила специальная архитектура искусственных нейронных сетей, известная как свёрточная нейронная сеть (СНС). Она предложена Яном Лекуном в 1988 году [1] и нацелена на распознавание изображений [2].

Существуют решения по распознаванию типа помещений/комнат. Например, в работе [3] предлагается подход к определению типа помещения с использованием эхолокация и СНС. Авторы используют преобразованные данные эхолокации, на которых затем обучают сеть.

© А. В. Смирнов, Д. Н. Степанов, 2018
© Институт программных систем имени А. К. Айламазяна РАН, 2018
© Программные системы: теория и приложения (дизайн), 2018

10.25209/2079-3316-2018-9-4-279-291

Также существуют решения по навигации внутри помещений с использованием монокулярной камеры и распознавания помещений с использованием СНС. В работе [4] рассматривается задача навигации мобильного робота в помещениях, в частности, обход препятствий. Авторы используют модифицированную сеть AlexNet [5]. По результатам распознавания сеть выдает одну из трёх команд управления: поворот налево, поворот направо и движение вперёд.

В работе [6] представлен метод навигации с использованием распознавания панорамного изображения типа «рыбий глаз». Авторы предлагают модифицированную структуру СНС, основанную на структуре, представленной в [7]. На выходе у сети — одно из примерных направлений движения (от 3 до 7 направлений). Попутно создаётся база данных панорамных изображений и проводятся тесты на точность определения направления.

В работах [8, 9] вначале выполнялась предварительная съемка внутренних элементов планировки здания с созданием базы данных из большого количества фотографий (более 170000 изображений в одной работе и более 600000 в другой) для обучения СНС, а затем на тестовых фотографиях распознавалось конкретное помещение, или же необходимо было определить команду для дальнейшего движения беспилотного аппарата, летающего внутри помещений. Точность распознавания составляла от 70 до 80 %.

В большинстве подходов навигации с использованием нейросетевого распознавания помещений или внутренних элементов планировки здания используется механизм сбора изображений помещения с последующим обучением сети на них. Предлагается схожий подход для навигации мобильного робота внутри помещений. В отличие от уже существующих решений, для распознавания помещений используется одноканальное полутоновое изображение, содержащее определённые черты (края, границы) исследуемого помещения, а также собственная структура свёрточной нейронной сети. В результате создаются предпосылки для решения задачи первичной локализации мобильного робота с последующим определением дальнейших действий и составлением стратегии движения.

1. Создание базы данных

Для распознавания внутренних элементов планировки здания с помощью СНС предлагается создать базу данных (БД) полутоновых изображений, содержащих определённые черты исследуемых элементов

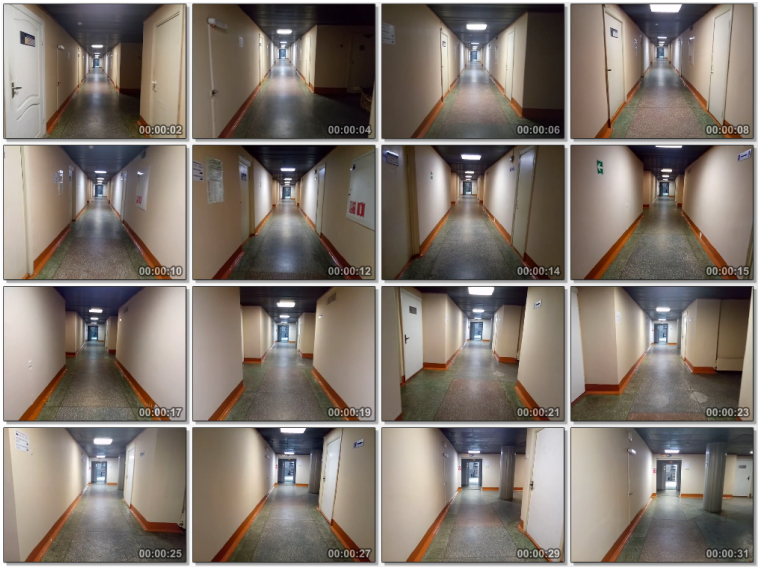

Рис. 1. Раскадровка видеофрагмента с движением по коридору

(края или границы). Использование полутоновых изображений с границами внутренних элементов планировки здания позволяет создать собирательный образ определённого типа помещения вне зависимости от конкретного здания. Создание БД заключается в покадровой обработке коротких видеофрагментов.

Для создания базы данных было записано 5 видеофрагментов длиной от 5 до 33 секунд с движением по коридору, 6 видеофрагментов длиной от 4 до 28 секунд с движением по лестнице, 16 видеофрагментов длинной от 3 до 8 секунд с движением рядом с различными дверьми и 11 видеофрагментов от 3 до 12 секунд с угловыми конструкциями стен помещения. Каждый из записанных видеофрагментов имел частоту кадров, равной 25, и разрешение 640×480 пикселей. Раскадровка одного из записанных фрагментов представлена на рис. 1.

1.1. Фильтрация изображения и выделение особых черт

Каждый видеофрагмент проходит покадровую обработку с применением библиотеки компьютерного зрения OpenCV [10]. Алгоритм обработки кадров состоит из нескольких шагов:

(*1*) изменение размера исходного кадра на 300×300 пикселей,

Рис. 2. Обработанный кадр видеофрагмента с движением по коридору

(*2*) применение фильтра edgePreservingFilter,

(*3*) перевод изображения из цветного в полутоновое,

(*4*) применение оператора Laplacian,

(*5*) изменение размера выходного изображения на 64×64 пикселя.

Первичное изменение размера уменьшает время фильтрации изображения. Перевод изображения из цветного в полутоновое и вторичное изменение размера уменьшают количество параметров, используемых в СНС. Фильтр edgePreservingFilter [**11**] осуществляет сглаживание, сохраняя детали на границах объекта, оператор Laplacian выделяет границы и определяется по следующей формуле:

$$Laplace(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$

В ходе экспериментов по выделению границ внутренних элементов планировки здания использовались различные комбинации фильтров. Для удаления шумов использовалась медианная фильтрация (medianBlur), Гауссово размытие (GaussianBlur). Однако, в отличии от edgePreservingFilter, данные методы обработки изображений размывают изображение без учёта границ. Также для выделения границ использовался оператор Кэнни (Canny) [**12**]. Для использования оператора Кэнни требуется вручную указывать значения двух параметров (верхняя и нижняя граница двойной пороговой фильтрации). Оптимальные значения сильно зависят от особенностей самого изображения, а вопрос адаптивного подбора таких параметров в данном исследовании не рассматривался, в связи с чем была выбрана комбинация фильтра edgePreservingFilter и оператора Laplacian для фильтрации изображений и выделения особых черт (границ/краёв) помещений. Результат обработки изображения показан на рис. 2.

Таким образом в автоматическом режиме генерируются изображения для БД с использованием видеофрагментов, содержащих

Таблица 1. Пример изображений в выборках

Номер и имя класса		Примеры изображений
0	angle	
1	coridor	
2	door	
3	stairs	

различные типы (классы) внутренних элементов планировки здания для создания обучающей и тренировочной выборок.

1.2. Разбиение полученных изображений на классы

После обработки видеофрагментов была создана база данных из 4 классов. Для каждого из классов обучающая выборка состояла из 900, а тренировочная — из 300 изображений размером 64×64 пикселя. Всего обучающая выборка состоит из 3600 изображений и тестовая — из 1200 изображений. Примеры изображений классов представлены в таблице 1.

2. Используемая структура СНС

Структура СНС, используемая в предложенном подходе, представлена в таблице 2.

Представленная СНС содержит 3 слоя свёртки с фильтром (ядром свёртки) размером 5×5 и методом обработки краёв исходной матрицы same (рис. 3), 4 слоя подвыборки (max pooling) и два полносвязных слоя. В качестве функции активации используется ReLU [13] и Softmax [14] для последнего полносвязного слоя.

Таблица 2. Структура СНС

Слой	Вход	Фильтр	Выход
свёртка (same)	64x64x1	5x5	64x64x32
подвыборка (max)	64x64x32	2x2	32x32x32
функция активации (ReLU)	32x32x32		32x32x32
свёртка (same)	32x32x32	5x5	32x32x32
подвыборка (max)	32x32x32	2x2	16x16x32
функция активации (ReLU)	16x16x32		16x16x32
свёртка (same)	16x16x32	5x5	16x16x64
подвыборка (max)	16x16x64	2x2	8x8x64
функция активации (ReLU)	8x8x64		8x8x64
подвыборка (max)	8x8x64	2x2	4x4x64
полносвязный	4x4x64		64
полносвязный	64		4
функция активации (Softmax)	4		4

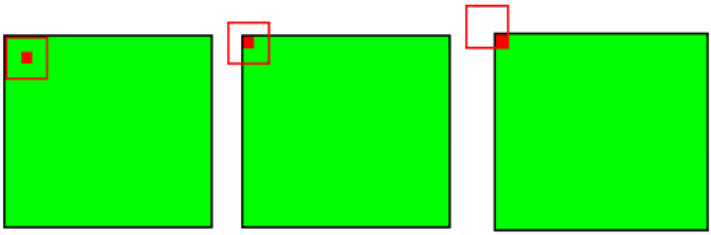

Рис. 3. Методы обработки краёв исходной матрицы. Слева направо — valid, same, ful

2.1. Используемый инструментарий

В качестве программной платформы для создания СНС, а также её обучения и проведения экспериментов использовалась библиотека tiny-dnn [15], которая является “header-only” библиотекой, не требует сборки и легко интегрируется в проекты. Библиотека содержит готовые реализации различных слоёв СНС, функций активации и алгоритмов оптимизации.

Разработка предложенного метода велась в системе блочно-параллельной обработки данных [16]. Реализованный в системе модульный принцип обработки информации позволяет формировать части программного кода в отдельные модули и использовать их независимо

Рис. 4. Разработанные схемы

друг от друга. Модуль может иметь от одного до нескольких каналов входа/выхода. Разработаны и использованы следующие модули:

ReadMultiImage — модуль чтения одного или нескольких изображений,

Resize_image — модуль изменения размера изображения,

EdgePreservingFilter — модуль применения одноимённого фильтра,

ColorTransform — модуль перевода изображения в разные цветовые пространства,

Laplacian — модуль применения оператора Лапласа,

ParseSave_image — модуль перевода значения пикселей изображений из формата $0 \ldots 255$ в $-1 \ldots 1$, передачи значений в вектор и сохранения в файл .bin,

ReadBin — модуль чтения вектора данных из формата .bin,

CNN_Tiny — универсальный модуль обучения, теста, распознавания сети на основе tiny-dnn.

Модули взаимодействуют друг с другом в соответствии с определёнными «задачами». «Задача» представляет собой специальный файл с описанием последовательности использования модулей, способом их связи и входными параметрами. С помощью графического интерфейса системы каждую «задачу» можно отобразить в виде схемы (рис. 4,

в которой над названием модуля отображается тип данных для входного канала, под названием — тип данных выходного канала).

Процесс предобработки заключается в следующем: запуск модуля ReadMultiImage для загрузки последовательно нескольких изображений. Далее у изображения изменяется размер в соответствии с предложенным ранее алгоритмом (модуль Resize_image). Затем происходит фильтрация и преобразование изображения в полутоновое (модули: EdgePreservingFilter -> ColorTransform -> Laplacian). После второго изменения размера происходит преобразование значений пикселей изображения из формата $0 \ldots 255$ в формат $-1 \ldots 1$, передача значений пикселей в вектор и сохранение в .bin файл, с помощью модуля ParseSave_image.

При обучении сети модуль ReadBin служит для чтения вектора данных, полученных ранее. Основной модуль сети CNN_Tiny содержит структуру СНС и может выполнять обучение, тест или распознавание в зависимости от введённых параметров. Выполнение теста сети отличается от обучения установкой параметра в модуле CNN_Tiny на автотестирование и создание таблицы точности.

Процесс распознавания единичного входного изображения отличается от процесса предобработки передачей обработанного изображения напрямую модулю нейронной сети с параметром на распознавание.

2.2. Результаты обучения и распознавания

Обучение СНС представленной выше структуры проводилось на CPU Intel® Core™ i5-3470 с частотой 3,20 GHz с использованием средств параллельной обработки TBB [17] и инструкций AVX [18]. Обучение составило 30 эпох со средним временем на эпоху около 22 с. Точность распознавания составила 96,83 %. Информация о количестве распознанных изображений по классам тестовой выборки представлена в таблице 3.

Названия столбцов и строк в таблице 3 соответствует названиям классов, представленных в таблице 1. Цифры в ячейках таблицы соответствуют количеству изображений тестовой выборки (всего 300 изображений на класс), которые сеть распознала и отнесла к определённому классу. Таким образом, чем больше число в ячейке, соответствующей пересечению строки и столбца одинаковым именем, тем точнее распознался тот или иной класс.

Таблица 3. Количество распознанных изображений тестовой выборки по классам

Класс	Распознанный класс			
	angle	coridor	door	stairs
angle	299	0	2	0
coridor	0	281	0	0
door	1	0	282	0
stairs	0	19	16	300

Заключение

Представленный метод показал результат успешного распознавания типов внутренних элементов планировки здания более 96 %. Процент распознавания можно улучшить, если разнообразить коллекцию обучающих изображений, дополнив ее съемками с камеры в разных точках коридоров, с разными углами и дверями. Использование данного метода может стать отправной точкой в решении задачи ориентации робота в начальный момент времени для дальнейшего составления списка команд движения робота. Например, если определено, что робот находится в коридоре, то можно отдать команду на движение вперёд. Если же робот стоит перед препятствием (например дверь), то следует отдать команду на разворот.

Дальнейшая работа заключается в расширении числа классов исследуемых помещений и числа изображений в обучающей и тестовой выборках. Кроме этого планируется использовать возможности GPU для распознавания и провести эксперименты с использованием реального прототипа мобильного робота.

Список литературы

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel. “Backpropagation applied to handwritten zip code recognition”, *Neural Computation*, **1**:4 (1989), pp. 541–551. URL ↑279

[2] Convolutional Neural Networks (LeNet) (дата обращения: 25.07.2018). URL ↑279

[3] Q. Song, C. Gu, R .Tan. “Deep Room Recognition Using Inaudible Echos”, *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, **2**:3 (2018), pp. 135-163. ↑279

[4] C. Liu, B. Zheng, C. Wang, Y. Zhao, S. Fu, H. Li. “CNN-based vision model for obstacle avoidance of mobile robot”, *MATEC Web of Conferences*, **139** (2017). doi ↑280

[5] A. Krizhevsky, I. Sutskever, G. E. Hinton. “ImageNet classification with deep convolutional neural networks”, *NIPS’12 Proceedings of the 25th International Conference on Neural Information Processing Systems*. V. 1, 2012, pp. 1097–1105. URL ↑280

[6] L. Ran, Y. Zhang, Q. Zhang, T. Yang. “Convolutional neural network-based robot navigation using uncalibrated spherical images”, *Sensors*, **17**:6 (2017). doi ↑280

[7] A. Giusti, J. Guzzi, D. Ciresan, F. L. He, J. P. Rodriguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. Di Caro. “A machine learning approach to visual perception of forest trails for mobile robots”, *IEEE Robotics and Automation Letters*, **1**:2 (2016), pp. 661-667. doi ↑280

[8] D. K. Kim, T. Chen. *Deep neural network for real-time autonomous indoor navigation*, 2015. URL ↑280

[9] F. Zhang, F. Duarte, R. Ma, D. Milioris, H. Lin, C. Ratti. *Indoor space recognition using Deep Convolutional Neural Network: A case study at MIT campus*, 2016. URL ↑280

[10] OpenCV (Open Source Computer Vision Library) (дата обращения: 25.07.2018). URL ↑281

[11] Non-Photorealistic Rendering using OpenCV (Python, C++) (дата обращения: 25.07.2018). URL ↑282

[12] J. Canny. “A computational approach to edge detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**:6 (1986), pp. 679–698. doi ↑282

[13] V. Nair, G Hinton. “Rectified linear units improve restricted Boltzmann machines”, *ICML’10 Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010, pp. 807–814. URL ↑283

[14] *Understand the softmax function in minutes* (дата обращения: 25.07.2018). URL ↑283

[15] T. Nomi, Tiny-dnn documentation, Oct 23, 2018. URL ↑284

[16] A. Kondratyev, I Tishchenko. “Concept of distributed processing system of images flow”, *Robot Intelligence Technology and Applications 4*, Advances in Intelligent Systems and Computing, vol. **447**, Springer, 2016, pp. 551–561. doi ↑284

[17] Threading Building Blocks (Intel® TBB) (дата обращения: 25.07.2018). URL ↑286

[18] ISA Extensions (дата обращения: 25.07.2018). URL ↑286

Поступила в редакцию 15.11.2018
Переработана 23.11.2018
Опубликована 17.12.2018

Рекомендовал к публикации *д.т.н. В. М. Хачумов*

Пример ссылки на эту публикацию:

А. В. Смирнов, Д. Н. Степанов. «Распознавание типа внутренних элементов планировки зданий в задачах первичной навигации мобильных роботов с использованием нейронных сетей». *Программные системы: теория и приложения*, 2018, **9**:4(39), с. 279–291.

10.25209/2079-3316-2018-9-4-279-291

http://psta.psiras.ru/read/psta2018_4_279-291.pdf

Об авторах:

Александр Владимирович Смирнов

М.н.с. Исследовательского центра мультипроцессорных систем ИПС им. А.К. Айламазяна РАН. Область научных интересов: компьютерное зрение, распознавание образов, визуальная навигация, нейронные сети, робототехника.

0000-0002-7104-1462

e-mail: asmirnov_1991@mail.ru

Дмитрий Николаевич Степанов

К.т.н., н.с. Исследовательского центра мультипроцессорных систем ИПС им. А.К. Айламазяна РАН. Область научных интересов: математическое моделирование, компьютерное зрение, распознавание образов, параллельное программирование, визуальная навигация, анализ данных.

0000-0003-2582-5757

e-mail: mitek1989@mail.ru

UDC 004.932.72'1

Alexander Smirnov, Dmitry Stepanov. *The use of convolutional neural networks for recognition of the type of premises using special features of the premises.*

ABSTRACT. The article proposes a method for recognizing a convolutional neural network (CNN) of the type / class of internal elements of a building layout using specific features of these elements, such as borders or edges. An algorithm for pre-processing of images is proposed to highlight the borders / edges in the image. Also a database is created with images of various types of interior elements of the building layout, such as a corridor, a door (doorway), corner structures and stairs. The article also discusses the own structure of the SNA, and presents data on the accuracy of recognition of various types of premises. The developed method is proposed to use for the primary navigation of mobile robots. (*In Russian*).

Key words and phrases: convolutional neural network, premises recognition, image filtering, block-parallel processing, feature extraction.

2010 *Mathematics Subject Classification*: 68T40; 68T45, 97P30

References

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel. "Backpropagation applied to handwritten zip code recognition", *Neural Computation*, **1**:4 (1989), pp. 541–551. ↑279

[2] Convolutional Neural Networks (LeNet) (accessed: 25.07.2018). ↑279

[3] Q. Song, C. Gu, R .Tan. "Deep Room Recognition Using Inaudible Echos", *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, **2**:3 (2018), pp. 135-163.↑279

[4] C. Liu, B. Zheng, C. Wang, Y. Zhao, S. Fu, H. Li. "CNN-based vision model for obstacle avoidance of mobile robot", *MATEC Web of Conferences*, **139** (2017). ↑280

[5] A. Krizhevsky, I. Sutskever, G. E. Hinton. "ImageNet classification with deep convolutional neural networks", *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*. V. 1, 2012, pp. 1097–1105. ↑280

[6] L. Ran, Y. Zhang, Q. Zhang, T. Yang. "Convolutional neural network-based robot navigation using uncalibrated spherical images", *Sensors*, **17**:6 (2017). ↑280

[7] A. Giusti, J. Guzzi, D. Ciresan, F. L. He, J. P. Rodriguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. Di Caro. "A machine learning approach to visual perception of forest trails for mobile robots", *IEEE Robotics and Automation Letters*, **1**:2 (2016), pp. 661-667. ↑280

[8] D. K. Kim, T. Chen. *Deep neural network for real-time autonomous indoor navigation*, 2015. ↑280

© A. V. SMIRNOV, D. N. STEPANOV, 2018
© AILAMAZYAN PROGRAM SYSTEMS INSTITUTE OF RAS, 2018
© PROGRAM SYSTEMS: THEORY AND APPLICATIONS (DESIGN), 2018

10.25209/2079-3316-2018-9-4-279-291

[9] F. Zhang, F. Duarte, R. Ma, D. Milioris, H. Lin, C. Ratti. *Indoor space recognition using Deep Convolutional Neural Network: A case study at MIT campus*, 2016. URL ↑280

[10] OpenCV (Open Source Computer Vision Library) (accessed: 25.07.2018). URL ↑281

[11] Non-Photorealistic Rendering using OpenCV (Python, C++) (accessed: 25.07.2018). URL ↑282

[12] J. Canny. "A computational approach to edge detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**:6 (1986), pp. 679–698. doi ↑282

[13] V. Nair, G Hinton. "Rectified linear units improve restricted Boltzmann machines", *ICML'10 Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010, pp. 807–814. URL ↑283

[14] *Understand the softmax function in minutes* (accessed: 25.07.2018). URL ↑283

[15] T. Nomi, Tiny-dnn documentation, Oct 23, 2018. URL ↑284

[16] A. Kondratyev, I Tishchenko. "Concept of distributed processing system of images flow", *Robot Intelligence Technology and Applications 4*, Advances in Intelligent Systems and Computing, vol. **447**, Springer, 2016, pp. 551–561. doi ↑284

[17] Threading Building Blocks (Intel® TBB) (accessed: 25.07.2018). URL ↑286

[18] ISA Extensions (accessed: 25.07.2018). URL ↑286

Sample citation of this publication:

Alexander Smirnov, Dmitry Stepanov. "The use of convolutional neural networks for recognition of the type of premises using special features of the premises". *Program Systems: Theory and Applications*, 2018, **9**:4(39), pp. 279–291. (*In Russian*). doi 10.25209/2079-3316-2018-9-4-279-291

URL `http://psta.psiras.ru/read/psta2018_4_279-291.pdf`