УДК 004.9

ПОДГОТОВКА АННОТИРОВАННЫХ ДАННЫХ ДЛЯ ОБУЧЕНИЯ АЛГОРИТМА РАСПОЗНАВАНИЯ

[1]Першина Ж.С., [1,2]Колкер А.Б., [1,2]Ощепкова С.А.

[1]ФГБОУ ВО «Новосибирский государственный технический университет», Новосибирск, e-mail: s0n4a@bk.ru;

[2]ФГБУ «Сибирский региональный научно-исследовательский гидрометеорологический институт», Новосибирск

В статье представлен аннотированный набор данных, предназначенный для обучения, валидации и тестирования алгоритмов семантической сегментации с определением экземпляров класса объектов на изображении (англ. instance semantic segmentation), реализованных на основе свёрточных нейронных сетей (англ. Convolutional neural network, CNN). Особенностью представленного набора данных является то, что он состоит из изображений реальных объектов и изображений объектов, полученных синтетическим путем. Сбор и генерация данных выполнена с изменением освещения, ракурса камеры относительно объектов сцены, положения и ориентации объектов в сцене, в том числе с учетом перекрытий. Аннотирование изображений реальных объектов в сцене выполнено с использованием разработанного инструмента полуавтоматического аннотирования данных. Аннотация синтетических данных получена в процессе генерации изображений объектов на основе технологии трассировки лучей POV-Ray. Оба способа позволяют создавать в кратчайшие сроки собственные качественно размеченные наборы данных. Приведены результаты оценки точности алгоритма семантической сегментации с определением экземпляров класса объектов на изображении, реализованного на основе модели Mask R-CNN, обученного с использованием созданного набора данных. Набор данных размещен в открытом доступе и может быть использован с целью проведения собственных исследований.

Ключевые слова: набор данных, сверточные нейронные сети, семантическая сегментация, технология трассировки лучей

PREPARATION OF ANNOTATED DATA FOR RECOGNITION ALGORITHM TRAINING

[1]Pershina Zh.S., [1,2]Kolker A.B., [1,2]Oschepkova S.A.

[1]Novosibirsk State Technical University, Novosibirsk, e-mail: s0n4a@bk.ru;

[2]FSBI SibNIGMI, Novosibirsk

The article presents annotated dataset designed for training, validation and testing of instance semantic objects' segmentation algorithms using convolutional neural networks. A feature of the presented dataset is the fact that it consists of real objects images and the images of objects which were synthetically obtained. The data was collected and generated in the different conditions of lighting, camera angle relative to scene objects, position and orientation of objects in the scene, with taking into account overlaps. Annotation or real objects images in the scene are carried out using the developed tool of automatic data annotation. The synthetic data annotation were obtained during the process of generating objects' images based on the POV-Ray ray tracing technology. Both methods make it possible to create your own qualitatively marked datasets as soon as possible. The paper presents the training and testing results of instance semantic segmentation algorithm, which was implemented on the basis of Mask R-CNN, are presented with using created dataset by model. The dataset is publicly available and can be used for your own research.

Keywords: dataset, convolutional neural networks, semantic segmentation, ray tracing technology

Семантический анализ изображений – одна из важных задач компьютерного зрения. Быстрое развитие подходов по распознаванию образов, основанных на свёрточных нейронных сетях (англ. convolutional neural networks, CNN) в совокупности с появлением крупномасштабных публичных наборов данных изображений, таких как PASCAL VOC [1], Cityscapes Dataset [2], CamVid [3], KITTY [4] и COCO [5], сделали возможным реализацию алгоритмов семантической сегментации объектов в сцене [6–8]. Однако для решения практических задач, в которых требуется высокая точность семантической сегментации, требования к качеству разметки возрастают и публичные наборы данных зачастую не удовлетворяют этому требованию. Другой причиной невозможности использования публичных данных может являться отсутствие в них необходимых для распознавания классов объектов. По этим причинам возникает задача создания собственного набора данных под специфику задачи. Создание собственного набора данных предполагает сбор данных и их аннотирование. Сбор данных может выполняться как путем синтетической генерации изображений объектов

при наличии их CAD-моделей, моделируя различные условия съемки (расположение источника освещения относительно камеры, расстояние от камеры до деталей в сцене, вращение и смещение камеры относительно сцены), так и посредством съемки реальных объектов. При этом аннотация синтетических данных формируется в процессе генерации изображений, а аннотирование изображений реальных объектов может быть выполнено ручным способом с использованием одного из следующих инструментов: Computer Vision Annotation Tool (CVAT) [9], LabelMe [10], Prodigy [11], VGG Image Annotation [12], RectLabel [13], Fluid Annotation [14, 15]. Использование данных инструментов предполагает разметку изображения путем выделения полигонов, содержащих объект, вручную и присвоение класса и/или уникального номера экземпляру объекта также вручную. При этом полигоны не позволяют учесть сквозные отверстия в объекте, что является источником шума при обучении и, как следствие, ухудшает точность распознавания. Также стоит отметить, что качество ручной разметки всецело зависит от качества работы аннотатора данных и требует значительных временных затрат. В связи с этим нами разработаны инструментальные средства, позволяющие выполнять разметку полуавтоматически, при этом не ухудшая качество разметки, а даже улучшая ее.

Цель исследования: исследование возможности использования синтетических данных, полученных в процессе генерации изображений объектов на основе технологии трассировки лучей POV-Ray, и её влияние на потерю точности сегментации объектов на реальных кадрах.

Набор данных

В качестве объектов использованы детали DIN стандарта: DIN1480 талреп крюк-кольцо (рис. 1, а), DIN82101 скоба такелажная (рис. 1, б), DIN580 рым-болт (рис. 1, в). CAD-модели этих объектов имеются в открытом доступе, а приобретение самих деталей не вызовет сложности (детали DIN стандарта могут быть приобретены в любом магазине крепежа и использованы для экспериментов с реальными объектами).

Набор данных состоит из 650 аннотированных изображений (204 искусственно сгенерированных изображений и 446 реальных изображений объектов), которые получены двумя способами:

1) генерация синтетических изображений объектов, которая выполнена при помощи технологии трассировки лучей POV-Ray;

2) сбор изображений реальных объектов с использованием камеры Intel RealSense D415.

Синтетические данные

Метод, использованный при генерации синтетических данных, описан в [16] и основан на технологии трассировки лучей POV-Ray [17]. Трассировка лучей – это процесс моделирования реального физического процесса поглощения и отражения света. Такой подход позволяет создавать реалистичные изображения объектов при различных условиях освещения. Каждый кадр имеет размеры 704×704 пикселя. Данный размер изображения обусловлен требованиями используемой нейросетевой модели Mask R-CNN [18]. Для обеспечения плавного масштабирования каждого элемента пиксельная кратность линейных размеров изображения должна составлять 64. Кратность 64 отображает верхний и нижний уровни 6 уровневой пирамиды FPN (2 * * 6 = 64) (англ. Feature Pyramid Network). Во избежание переобучения и повышения качества распознавания объектов в качестве фона использованы случайные изображения, полученные из кадров произвольного видеопотока, приведенного к размеру с необходимой кратностью. Пример полученных изображений представлен на рис. 2.

а) *б)* *в)*

Рис. 1. Объекты: а) талреп крюк-кольцо DIN1480; б) скоба такелажная DIN82101; в) рым-болт DIN580

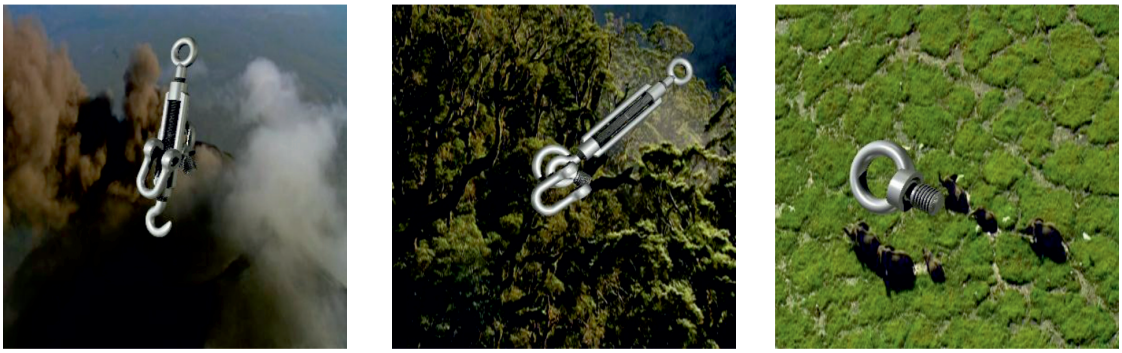

Рис. 2. Синтетические изображения с рандомизированным фоном

Реальные данные

Изображения реальных объектов получены с использованием Intel RealSense D415, и их размер составляет 640×480 пикселей. На рис. 3 представлен пример реальных объектов в наблюдаемой сцене.

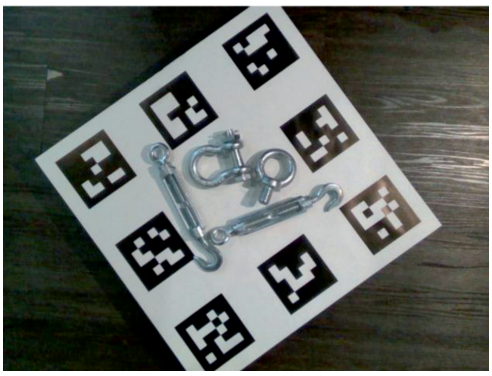

Рис. 3. Пример изображений реальных объектов

При сборе реальных данных учтены следующие факторы, влияющие на качество обучения: освещение, фон и степень перекрытия объектов друг относительно друга. Таким образом в наборе данных присутствуют различные сцены с реальными объектами на разнообразном фоне (рис. 4).

Аннотирование изображений реальных объектов выполнено как ручным способом с использованием Computer Vision Annotation Tool (CVAT) [19], так и с использованием разработанного инструмента полуавтоматического аннотирования данных. Полуавтоматическое аннотирование данных реализовано путем проекции на последовательность кадров трехмерных CAD-моделей объектов сцены, положение и ориентация которых соответствуют положению и ориентации некоторой предварительно определенной плоской поверхности сцены, содержащей уникальные по своей структуре маркеры (ArUco, QR и пр.), которые позволят оценить смещение и поворот сцены и объектов в сцене относительно опорного кадра. Последующая проекция 3D точек с учетом полученного смещения и поворота CAD-моделей на текущий кадр позволяет получить пиксельные координаты масок и присвоить класс объекта и уникальный номер экземпляру класса на изображении (рис. 5), после этого выполнить сохранение этих данных в файл аннотации.

а) *б)* *в)*

Рис. 4. Усложненные реальные кадры: а) без перекрытий; б) с частичными перекрытиями; в) с «шумом» в виде дополнительных объектов, не участвующих в обучении

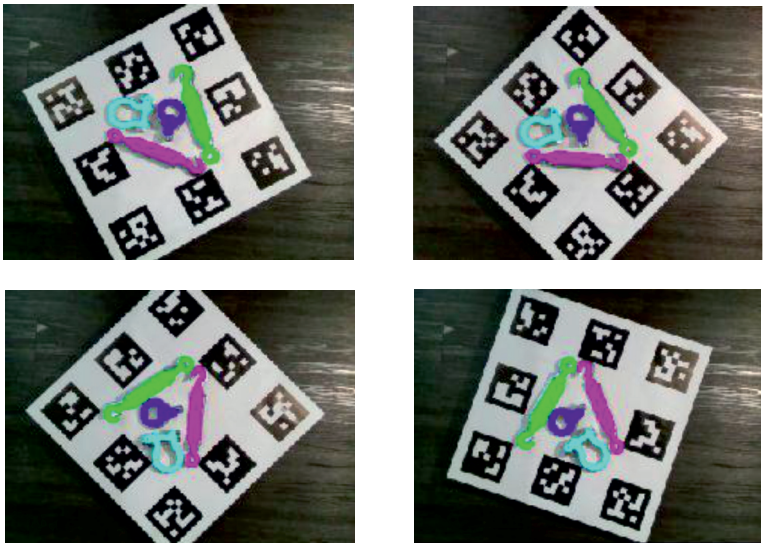

Рис. 5. Примеры размеченных изображений реальных объектов с использованием инструмента полуавтоматического аннотирования: зеленый: класс – талреп крюк-кольцо, id – 1; розовый: класс – талреп крюк-кольцо, id – 2; голубой: класс – скоба такелажная, id – 1; синий: класс – рым-болт, id – 1

Структура файла аннотации

Для описания аннотации выбран формат аннотирования COCO, упакованный в контейнер json.

Файл аннотации кадров содержит следующие записи:

{
"categories": [{"id": int, "name": str, "color": str}],
"images": [{"id": int, "width": int, "height": int, "file_name": str, "path": str}],
"annotations": [{"id": int, "image_id": int, "category_id": int, "width": int, "height": int, "area": int,
"segmentation": [], "bbox": [], "color": str, "iscrowd": int}]
}

Разделы файла аннотации должны содержать следующие поля:

1. Раздел “categories” содержит:
"Id" – номер класса для каждого объекта;
"Name" – название класса объекта;
"color" – цвет для данного класса объектов.
2. Раздел "images" содержит список параметров для каждого кадра:
"Id" – номер кадра;
“Width” – ширина кадра;
"Height" – высота кадра;
"file_name" – название файла (снимка), аннотация которого приводится;
"Path" – путь к аннотируемому изображению.
3. Раздел "annotations" содержит список параметров для каждого объекта в кадре:
"id" – номер экземпляра класса в кадре;
"image_id" – номер изображения;
"category_id" – номер класса аннотированного объекта;
“Width” – ширина кадра;
"Height" – высота кадра;
“Area” – территория кадра, занимаемая объектом;
"Segmentation" – полигон, содержащий объект в кадре;
"Bbox" – значения, определяющие углы ограничительной рамки;
"color" – цвет детали;
“Iscrowd” – значение 0 или 1 в зависимости от того, является ли маска объекта в кадре полигоном (многоугольником) или несжатым RLE (множеством).

Экспериментальное исследование

Для определения положения объекта в кадре используется модель Mask R-CNN [1], которая позволяет получить семантическую сегментацию объектов в кадре, а также сегментацию отдельных экземпляров одного класса на изображении. В качестве метрики для оценки точности работы алгоритма использована IoU (англ. Interception Over a Union) (*). Количество пикселей в пересечении целевой и полученной в результате семантической сегментации масок, деленное на общее количество пикселей, присутствующих в обеих масках.

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \qquad (*)$$

где TP (англ. True Positive) – верно принятое, FP (англ. False Positive) – неверно принятое (ошибка первого рода), FN (англ. False Negative) – неверно отвергнутое (ошибка второго рода). Алгоритм обучался на представленных данных, а затем проводилась оценка точности семантической сегментации объектов на тестовой выборке, результаты представлены в таблице.

Результаты тестирования

Точность, посчитанная с использованием весов, обученных на:	Синтетические данные	Реальные кадры
синтетических данных	IoU: 0.8953	IoU: 0.5460
реальных кадрах	IoU: 0.6921	IoU: 0.6054
смешанных данных	IoU: 0.8929	IoU: 0.6761

При сегментации объектов на реальных снимках были получены значения метрик IoU = 0,6054 при обучении модели на реальных снимках, и IoU = 0,6761 при обучении алгоритма на смешанном наборе данных, содержащем 10 % реальных снимков от общего числа кадров. То есть расширение обучающей выборки из реальных снимков, путем добавления в неё синтетических кадров, позволило получить сегментацию объектов на реальных кадрах без потери точности, но с уменьшением трудозатрат по подготовке данных.

Результаты

Набор данных размещен в публичный доступ по ссылке [20]. Приведенные материалы могут быть использованы в качестве эталонного с целью проведения собственных исследований.

Заключение

В статье приводится описание набора аннотированных данных, состоящего из изображений, сгенерированных синтетически, и снимков реальных объектов, полученных с использованием камеры Intel RealSense D415 в различных условиях. Полученный набор данных применим для обучения алгоритмов семантической сегментации, реализованных на основе свёрточных нейронных сетей. Проведено обучение и тестирование модели Mask R-CNN с использованием представленных данных. По результатам экспериментов видно, что использование синтетических данных, полученных в процессе генерации изображений объектов на основе технологии трассировки лучей POV-Ray, позволило получить сегментацию объектов на реальных кадрах без потери точности, но с уменьшением трудозатрат по подготовке данных.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-58-76003.

Список литературы

1. Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, Alan Yuille. The Role of Context for Object Detection and Semantic Segmentation in the Wild. IEEE Xplore, 2014. [Electronic resource]. URL: https://ieeexplore.ieee.org/document/6909514/citations#citations (date of access: 11.11.2020).

2. Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding // arXiv, 2016. [Electronic resource]. URL: https://arxiv.org/pdf/1604.01685.pdf (date of access: 11.11.2020).

3. Gabriel J. Brostow, Julien Fauqueur, Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters. 2009. Vol. 30. no. 2. P. 88–97.

4. Hassan Alhaija, Siva Mustikovela, Lars Mescheder, Andreas Geiger, Carsten Rother. Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes // arXiv, 2018. [Electronic resource]. URL: https://arxiv.org/pdf/1708.01566.pdf (date of access: 11.11.2020).

5. Fleet D., Pajdla T., Schiele B., Tuytelaars T. Microsoft COCO: Common Objects in Context // arXiv, 2014. [Electronic resource]. URL: https://arxiv.org/pdf/1405.0312.pdf (date of access: 11.11.2020).

6. Ruiz-del-Solar J., Loncomilla P., Soto N. A Survey on Deep Learning Methods for Robot Vision // arXiv, 2018. [Electronic resource]. URL: https://arxiv.org/ftp/arxiv/papers/1803/1803.10862.pdf (date of access: 11.11.2020).

7. A Nguyen. Scene Understanding for Autonomous Manipulation with Deep Learning // arXiv, 2019. [Electronic resource]. URL: https://arxiv.org/pdf/1903.09761.pdf (date of access: 11.11.2020).

8. Mallick, Arijit, del Pobil, Angel P., Cervera, Enric. Deep Learning based Object Recognition for Robot picking task // 1Library, 2018. [Electronic resource]. URL: https://1library.co/document/q7w1ovkz-deep-learning-for-object-recognition-in-picking-tasks.html#pdf-content (date of access: 11.11.2020).

9. GitHub – openvinotoolkit/cvat: Powerful and efficient Computer Vision Annotation Tool (CVAT). [Electronic resource]. URL: https://github.com/openvinotoolkit/cvat (date of access: 11.11.2020).

10. GitHub – wkentaro/labelme: Image Polygonal Annotation with Python (polygon, rectangle, circle, line, point and im-

age-level flag annotation). [Electronic resource]. URL: https://github.com/wkentaro/labelme (date of access: 11.11.2020).

11. Prodigy · An annotation tool for AI, Machine Learning & NLP. [Electronic resource]. URL: https://prodi.gy/ (date of access: 11.11.2020).

12. GitLab – Files · master · Visual Geometry Group / via GitLab. [Electronic resource]. URL: https://gitlab.com/vgg/via/tree/master (date of access: 11.11.2020).

13. GitHub – ryouchinsa/Rectlabel-support: RectLabel – An image annotation tool to label images for bounding box object detection and segmentation. [Electronic resource]. URL: https://github.com/ryouchinsa/Rectlabel-support (date of access: 11.11.2020).

14. M Andriluka, JRR Uijlings, V Ferrari, Fluid annotation: a human-machine collaboration interface for full image annotation // arXive, 2018. [Electronic resource]. URL: https://arxiv.org/pdf/1806.07527.pdf (date of access: 11.11.2020).

15. Google AI Blog: Fluid Annotation: An Exploratory Machine Learning; Powered Interface for Faster Image Annotation. [Electronic resource]. URL: https://ai.googleblog.com/2018/10/fluid-annotation-exploratory-machine.html (date of access: 11.11.2020).

16. Kolker A., Oshchepkova S., Pershina Z., Dimitrov L., Ivanov V., Rashid A. and Bdiwi M. The Ray Tracing based Tool for Generation Artificial Images and Neural Network Training. In Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. 2020. Vol. 3. P. 257–264.

17. POV-Ray – The Persistence of Vision Raytracer. [Electronic resource]. URL: http://www.povray.org (date of access: 11.11.2020).

18. Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick. Mask R-CNN // arXiv, 2017. [Electronic resource]. URL: https://arxiv.org/pdf/1703.06870.pdf (date of access: 11.11.2020).

19. GitHub – openvinotoolkit/cvat: Powerful and efficient Computer Vision Annotation Tool (CVAT). [Electronic resource]. URL: https://github.com/openvinotoolkit/cvat (date of access: 11.11.2020).

20. GitHub – ICUERA/Bencmark_ICU. [Electronic resource]. URL: https://github.com/ICUERA/Bencmark_ICU (date of access: 11.11.2020).