

## СЕГМЕНТАЦИЯ ОБЪЕКТОВ С ФУНКЦИЕЙ ДООБУЧЕНИЯ

И. Д. НЕНАХОВ<sup>1\*</sup>, К. АРТЕМОВ<sup>1</sup>, С. ЗАБИХИФАР<sup>2</sup>,  
А. Н. СЕМОЧКИН<sup>2</sup>, С. А. КОЛЮБИН<sup>1</sup><sup>1</sup>Университет ИТМО, Санкт-Петербург, Россия  
\*ivdmne@gmail.com<sup>2</sup>Сбербанк, Москва, Россия

**Аннотация.** Рассматриваются способы расширения набора распознаваемых классов объектов для задачи их сегментирования, где необходимо построить маску объекта, а также узнать его класс. Для первой задачи использованы методы, не зависящие от классов предметов и являющиеся наиболее устойчивыми к изменениям формы; для второй задачи проанализированы методы, основанные на итеративном обучении (iterative learning), и методы метрического обучения (metric learning). Вторым подходом выбран в качестве основного, и для него протестированы различные архитектуры нейронных сетей. Проведена классификация объектов с использованием алгоритма  $k$  ближайших соседей. В качестве набора данных для обучения нейронной сети использован набор COIL-100, а после обученная модель протестирована на собственном наборе данных. Проведенные эксперименты показывают, что используемый метод позволяет обрабатывать 7-8 изображений в секунду на видеокарте GTX 1050 ti с 4 Гбайт видеопамяти с точностью классификации в 99 %.

**Ключевые слова:** метрическое обучение, итеративное обучение, сегментация, классификация, сверточные нейронные сети, робототехника, распознавание изображений

**Ссылка для цитирования:** Ненахов И. Д., Артемов К., Забихифар С., Семочкин А. Н., Колюбин С. А. Сегментация объектов с функцией дообучения // Изв. вузов. Приборостроение. 2022. Т. 65, № 3. С. 194—203. DOI: 10.17586/0021-3454-2022-65-3-194-203.

## OBJECTS SEGMENTATION WITH RETRAINING FUNCTION

I. D. Nenakhov<sup>1\*</sup>, K. Artemov<sup>1</sup>, S. Zabihifar<sup>2</sup>,  
A. N. Semochkin<sup>2</sup>, S. A. Kolyubin<sup>1</sup><sup>1</sup>ITMO University, St. Petersburg, Russia Россия  
\*ivdmne@gmail.com<sup>2</sup>Sberbank, Moscow, Russia

**Abstract.** Ways to expand the set of recognized object classes for the task of segmenting them, where it is necessary to build an object mask, as well as to find out its class, are considered. For the first task, methods that do not depend on the classes of subjects and are the most resistant to shape changes were used; for the second task, methods based on iterative learning and methods of metric learning are analyzed. The second approach is chosen as the main one, and various neural network architectures are tested for it. The classification of objects using the  $k$  nearest neighbors algorithm is carried out. The COIL-100 set is used as a data set for training a neural network, and after that the trained model was tested on its own data set. The experiments show that the method used allows processing 7-8 images per second on a GTX 1050 ti graphics card with 4 GB of video memory with a classification accuracy of 99%.

**Keywords:** metric learning, iterative learning, segmentation, classification, convolutional neural networks, robotics, image recognition

**For citation:** Nenakhov I. D., Artemov K., Zabihifar S., Semochkin A. N., Kolyubin S. A. Objects segmentation with retraining function. *Journal of Instrument Engineering*. 2022. Vol. 65, N 3. P. 194—203 (in Russian). DOI: 10.17586/0021-3454-2022-65-3-194-203.

**Введение.** Распознавание образов является одним из ключевых навыков человека для восприятия окружающего мира. Аналогично, роботу для имитации человека необходимо понимать, что его окружает. Как для человека основным органом чувств является зрение, так и

для робота основная часть информации о мире поступает через видеокамеры в виде двухмерных изображений. Однако недостаточно только получать данные с камер, их необходимо надлежащим образом обрабатывать.

Сегментация изображений — одна из основных задач в компьютерном зрении. Данная задача, как и большинство задач компьютерного зрения, сегодня решается с помощью алгоритмов, основанных на сверточных нейронных сетях (CNN), активное развитие которых отмечается в последнее десятилетие с достижением высоких показателей производительности компьютеров и, в частности видеокарт.

Компьютерное зрение в гуманоидной робототехнике имеет несколько отличительных характеристик и задач, отделяющих его от других областей компьютерного зрения, основанных на глубоком обучении. Автономный робот должен быть способен эффективно взаимодействовать со средой, предназначенной для человека, в частности, различать предметы в изменяющемся окружении и быть способным итеративно обучаться распознаванию объектов.

В настоящей статье рассматриваются различные методы итеративного обучения и исследуется методика метрического обучения, а также предлагается алгоритм, минимизирующий участие человека в процессе адаптации к новым классам объектов.

**Распознавание образов.** *Основные архитектуры сверточных нейронных сетей.* Операция свертки заключается в применении скользящего фильтра (как правило, малого размера) по изображению. Сама по себе свертка применима для нахождения примитивов: линий, углов, точек и т.д. Однако если использовать свертку не на исходном изображении, а на результате свертки этого изображения с каким-либо фильтром, то можно находить более сложные образы.

Первой удачной архитектурой сверточной нейронной сети является Alexnet [1], в 2013 г. превзошедшая классические алгоритмы в соревновании по классификации изображений на наборе данных ImageNet. Наряду со сверточными слоями в Alexnet применяются слой пулинга и полносвязный слой. Первый уменьшает размер карты признаков, деля тензор на сетку и выбирая среднее или максимальное его значение в элементах каждой ячейки. Полносвязный слой служит для классификации полученных высокоуровневых признаков.

Впоследствии нейронные сети стали все более глубокими. В очень глубоких нейронных сетях наблюдается проблема угасания градиента по мере его прохождения во время обновления весов. Нововведением архитектуры ResNet (2015) [2] является добавление обходных соединений. На определенных слоях создается соединение, пропускающее сигнал через несколько следующих слоев с последующей конкатенацией.

С увеличением глубины сети возрастает как вычислительная сложность, так и объем занимаемой памяти. Архитектура MobileNet [3] создавалась как компактная версия глубокой нейронной сети. В такой архитектуре для уменьшения пространственных размеров вектора используются блоки со свертками с шагом 2, а не пулинг, как применялось прежде. Также отличительной особенностью MobileNet является малый объем занимаемой памяти (16 Мбайт).

*Сегментация изображений.* Описанные выше архитектуры сверточных сетей применяются не только для классификации, но и служат остовом (backbone) моделей для остальных задач, таких как детектирование и сегментация. Рассмотрим подробнее задачу сегментации изображений.

При семантической сегментации ставится задача каждому пикселу входного изображения присвоить класс объекта, к которому он относится. При этом если два объекта одного класса соприкасаются, то на выходе модели нельзя определить их границу. Задача сегментации объектов (instance segmentation) решает эту проблему. Наряду с классом объекта необходимо определить, к какому объекту относится пиксел.

Наиболее успешное и популярное решение в задаче сегментации объектов — Mask R-CNN [4] является модификацией двухфазового (two-stage) детектора объектов Faster R-CNN [5]. На первом этапе изображение обрабатывается backbone-сетью. Далее берутся результаты с различных слоев этой сети и подаются в сеть для предсказания регионов (Region Proposal Network — RPN), которая выделяет области, где может содержаться объект. На этапе RoIAlign для каждого региона выделяется соответствующая ему карта признаков из карты признаков исходного изображения. Каждый найденный регион затем классифицируется полносвязными слоями и при наличии объекта уточняются границы региона и осуществляется поиск его маски. Эти три этапа производятся параллельно на признаках, полученных backbone-моделью. Структура модели Mask R-CNN приведена на рис. 1.

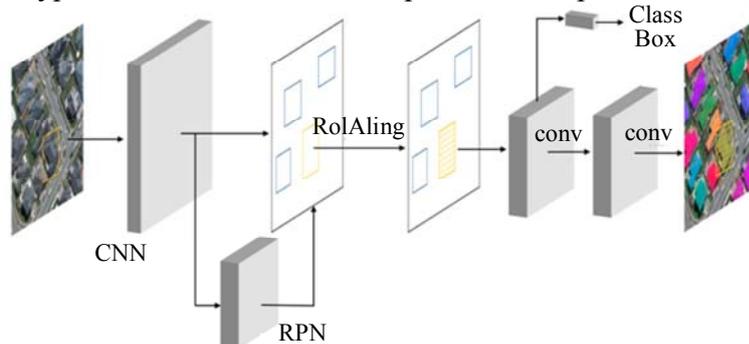


Рис. 1

**Адаптация классификации объектов к расширению набора классов.** Поставленную задачу можно сформулировать следующим образом. Пусть имеется исходное конечное множество объектов  $N_x = \{n_{x1}, n_{x2}, \dots, n_{xn}\}$  и множество соответствующих объектам классов  $N_y = \{n_{y1}, n_{y2}, \dots, n_{yk}\}$ . Также имеется счетное множество  $M_x = \{m_{x1}, m_{x2}, \dots\}$  дополнительных заранее не определенных объектов с соответствующими метками классов  $M_y = \{m_{y1}, m_{y2}, \dots\}$ , мощность которого также заранее не известна, но может быть оценена сверху как  $m^*$ . Множества  $N_y$  и  $M_y$  не пересекаются,  $N \cap M = \emptyset$ . Изначально алгоритм должен быть способен классифицировать объекты из множества классов  $C_0 \subset N_y$ . При этом в процессе работы алгоритма в явном виде в качестве новых обучающих данных могут выступать объекты, относящиеся к произвольному классу из множества  $M_y$ , требующие запуска процесса адаптации алгоритма. Следовательно, на каждой  $i$ -й итерации обучения алгоритм должен адаптироваться и расширять множество классифицируемых классов  $C_i = C_{i-1} \cup \{m_{yi}\}$ . В конечном счете требуется решить задачу классификации объектов из множества  $N \cup M$  максимальной мощностью  $(n+m^*)$  с точностью не ниже заданной:

$$A = \frac{T_p + T_n}{k + m^*},$$

где  $T_p$  — количество истинно положительных прогнозов,  $T_n$  — количество истинно отрицательных прогнозов, а  $(k + m^*)$  — количество объектов в выборке.

В настоящее время существует две области машинного обучения, способные решить данную задачу: итеративное обучение (iterative learning) и метрическое обучение (metric learning).

**Итеративное обучение.** Процесс обучения состоит в определении оптимальных параметров модели. Для каждого обучающего примера оптимальные параметры различаются, и цель обучения — найти такие параметры, которые позволят получить удовлетворительное решение на всем множестве обучающих данных. Для этого при обучении данные выбираются в случайном порядке. Но когда данные поступают последовательно, оптимальные параметры, найденные в текущей итерации, при последующих итерациях сойдутся к другим значениям. Таким образом, модель потеряет способность распознавать объекты предшествующих итераций. Этот эффект называется катастрофическим забыванием.

В сверточных нейронных сетях разработано множество методов итеративного обучения, направленных на преодоление катастрофического забывания. Их можно разделить на три вида. Первые — регуляризационные подходы [6, 7], ограничивающие возможность изменения наиболее „важных“ весов; вторые — архитектурные подходы [8, 9], которые основываются на изменении архитектуры после каждого обученного класса; третьи [10] — подходы, никак не влияющие на параметры сети, но использующие часть данных от прежних классов для „напоминания“ в последующих итерациях обучения. Также существуют работы [11], в которых рассматривается комбинация этих методик. Однако применение методик итеративного обучения сопряжено с достаточно длительным временем обучения или необходимостью изменять модель при каждой итерации.

*Метрическое обучение.* Задача метрического обучения — построить такой экстрактор признаков, чтобы векторы, соответствующие данным одного класса, находились близко друг к другу в пространстве признаков, а векторы, соответствующие данным разных классов, — далеко друг от друга. При этом в экстракторе не содержится информации о том, какому именно классу принадлежат входные данные.

Экстрактор признаков можно представить в виде функции, которая отображает входное изображение на пространство признаков фиксированной размерности —  $F: R^{whc} \rightarrow R^z$ , где  $w, h, c$  — размерность изображений,  $z$  — размерность итогового пространства признаков. Таким образом, каждому изображению ставится в соответствие вектор  $f \in R^z$ .

Для обучения такого экстрактора разработана функция потерь триплета (triplet loss) [12]

$$\text{Loss} = \sum_{i=1}^{k+m} \left[ \|f_i^a - f_i^p\|_2^2 - \|f_i^a - f_i^n\|_2^2 + \alpha \right],$$

принимая на вход тройку векторов: опорный  $f_i^a$ , положительный  $f_i^p$  и отрицательный  $f_i^n$ . Расстояние от опорного вектора до положительного должно быть меньше, чем от опорного до отрицательного как минимум на величину  $\alpha$ . При этом за положительный принимается вектор, класс которого совпадает с классом опорного, а за отрицательный — вектор, класс которого не совпадает с классом опорного. Далее необходимо сохранить векторы, представляющие изображения набора данных, и впоследствии производить классификацию новых изображений с помощью алгоритма  $k$  ближайших соседей ( $k$ -Nearest Neighbors — kNN).

Метод  $k$  ближайших соседей — один из наиболее тривиальных алгоритмов машинного обучения для классификации и регрессии. Изначально сохраняются точки обучающей выборки в пространстве признаков. Затем задается параметр  $k$ , указывающий на количество соседей, среди которых будет проводиться классификация. Для каждого тестового объекта находят  $k$  ближайших соседей в пространстве признаков и присваивается класс, наиболее часто встречающийся среди найденных соседей.

**Реализация процесса сегментации с возможностью дообучения.** *Описание алгоритма.* Так как алгоритмы метрического обучения позволяют быстро добавлять новые объекты, то было решено использовать обученный по данной методике экстрактор признаков и алгоритм kNN для классификации объектов, а для сегментации — Mask R-CNN. При этом экстрактор признаков, сеть для прогноза регионов и голова сети для определения маски предмета (mask head) в модели Mask R-CNN при обучении не имеют информации о классах предметов, поэтому возможно использовать эти компоненты для сегментации изображения без классификации. Найденные объекты после удаления фона подаются в экстрактор признаков и с помощью алгоритма kNN производится их классификация.

*Тестирование алгоритма.* Для решения задачи классификации были протестированы несколько архитектур экстракторов признаков: ResNet50, MobileNetV3\_small, MobileNetV3\_large.

Архитектура ResNet была выбрана, так как является одной из современных (state-of-the-art) моделей из набора данных ImageNet. Архитектуры MobileNet — легковесные модели, в некоторых задачах склонные к лучшей обобщающей способности, чем громоздкие аналоги.

Также имеет значение размерность выходного вектора — чем он больше, тем больший объем информации можно закодировать им и тем точнее будет классификация. Однако вместе с ростом размерности увеличивается размер файла, в котором хранятся ранее полученные признаки, а также увеличивается объем оперативной памяти во время работы программы.

Каждый из используемых экстракторов признаков формирует вектор заданной размерности. Для регулирования размерности после экстрактора признаков был добавлен один полносвязный слой с количеством нейронов, равным желаемой размерности вектора. В данном эксперименте были протестированы векторы следующих размерностей: 16, 32, 64 и 128 элементов.

Для тестирования был выбран набор данных COIL-100 [13]. Примеры объектов показаны на рис. 2. Набор данных содержит 100 объектов по 72 изображения на каждый. Для обучения были выбраны первые 50 объектов из набора данных. Обучение проводилось на протяжении 10 эпох. Для тестирования использовались оставшиеся 50 объектов. Работа алгоритмов оценивалась с помощью видеокарты Nvidia RTX 3080. Результаты обучения различных архитектур приведены в таблице.



Рис. 2

Архитектура	Размерность вектора	Точность, %	Время обучения, с	Время обработки одного изображения, мс
Resnet50	16	97,9	293	10
MobileNetV3 small	16	96,9	303	8
MobileNetV3 large	16	98,1	321	11
Resnet50	32	98,8	409	10
MobileNetV3 small	32	98,7	307	8
MobileNetV3 large	32	99,1	321	10
Resnet50	64	99,1	413	10
MobileNetV3 small	64	99,2	309	8
MobileNetV3 large	64	99,5	324	10
Resnet50	128	99,4	418	10
MobileNetV3 small	128	99,2	312	8
MobileNetV3 large	128	99,5	327	10
Resnet50	256	99,6	426	10
MobileNetV3 small	256	99,3	319	8
MobileNetV3 large	256	99,6	333	10
Resnet50	512	99,4	435	10
MobileNetV3 small	512	99,4	331	9
MobileNetV3 large	512	99,6	347	10

Как видно из таблицы, более легковесные архитектуры немного превосходят ResNet по точности и значительно по скорости обучения и времени обработки. Архитектура ResNet по сравнению в MobileNet имеет большее число параметров. Следовательно, модель может „запомнить“ больше обучающих примеров вместо выведения закономерностей, тем самым более склонна к переобучению. Этим объясняется несколько меньшая точность ResNet по сравнению с MobileNet.

Размерность пространства признаков оказывает влияние как на точность и скорость классификации, так и на объем памяти для хранения признаков. При увеличении размерности модель имеет больше способов разнести в пространстве точки разных классов, что повышает точность. В то же время расчет расстояния между двумя точками в таком пространстве будет занимать больше времени. Также кратно возрастает объем памяти для хранения векторов признаков.

Проецирование векторов признаков из 128-мерного пространства на плоскость с помощью алгоритма снижения размерности Umap представлено на рис. 3. Визуализация отображает точки, соответствующие векторам тестовой части (50 объектов) набора данных. Точки, соответствующие одному классу, сгруппированы друг с другом, тогда как точки разных классов находятся на значительном расстоянии друг от друга практически для всех классов объектов.

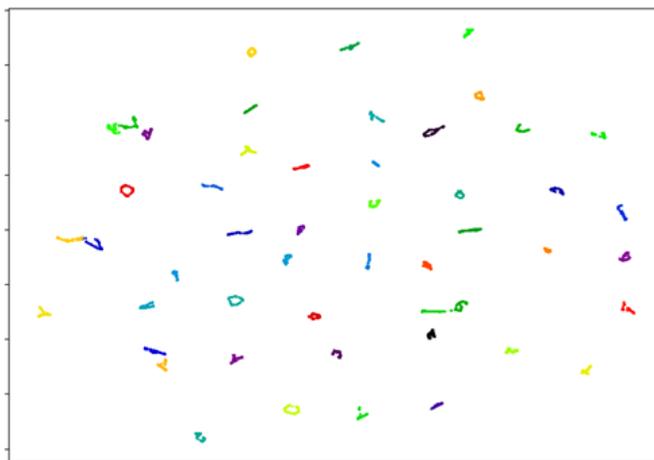


Рис. 3

*Тестирование алгоритма на реальных объектах.* После тестирования алгоритма на заранее подготовленном наборе данных был проведен эксперимент с использованием реальных объектов: кружка (cup), коробка (box), кубик (cube), компьютерная мышь (mouse) и антисептик (cleaner).

Для каждого объекта было снято видео длительностью 15 с, показывающее объект с разных сторон. Далее изображения были разделены на обучающую и тестовую части в соотношении 70 на 30. Обучающие 70 % изображений были использованы для получения векторов в пространстве признаков и обучения алгоритма  $k$  ближайших соседей, а оставшиеся 30 % — для оценки классификации. Точность обученной системы составила 98 %.

В то время как обученные классы объектов модель определяет с удовлетворительной точностью, в базовой конфигурации она часто распознает незнакомые предметы как обученные, т.е. имеют место ложно-положительные срабатывания. В этом случае необходимо фильтровать результаты классификации в соответствии с процентным соотношением количества объектов спрогнозированного класса среди  $k$  ближайших соседей, чтобы исключить объекты, находящиеся между кластерами известных классов. При тестировании алгоритма был выбран порог в 80 %. Также могут появиться объекты, все ближайшие соседи которых принадлежат к одному классу, но расстояние до первого из них велико. Тогда считается, что объект не принадлежит ни к одному известному классу. При тестировании был выбран порог для

косинусного расстояния в 0,2. Примеры сегментации для объектов из набора данных представлены на рис. 4.

Результаты 2D визуализации обучающих данных в пространстве признаков представлены на рис. 5. Можно заметить, что за редким исключением репрезентации различных объектов явно разделены. Используя видеокарту Nvidia GTX 1050ti с 4 Гбайт видеопамяти, алгоритм способен обрабатывать 7-8 изображений в секунду.

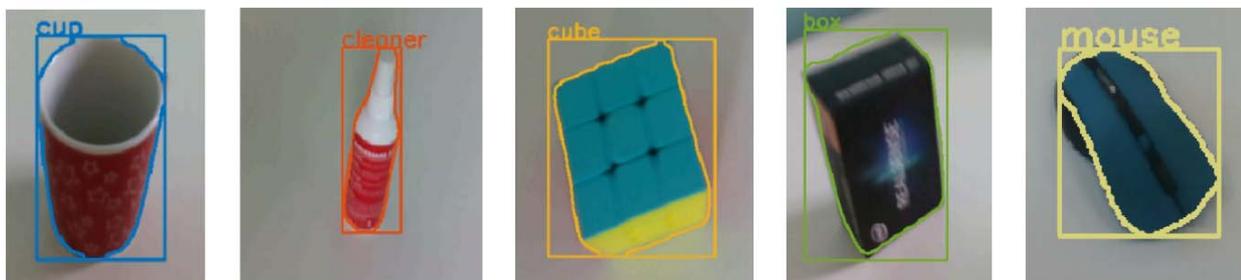


Рис. 4

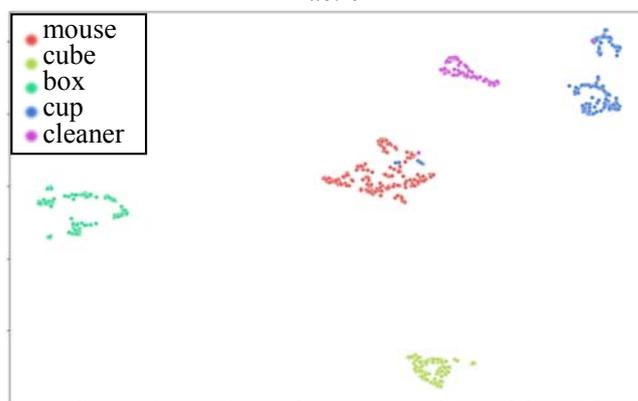


Рис. 5

При реализации данного алгоритма были использованы фреймворки Detectron2 [14] и PyTorch [15] и библиотека Pytorch Metric learning [16].

*Алгоритм автоматического дообучения.* Ключевой особенностью робототехники является минимизация участия человека в рабочем процессе. При поступлении объектов новых классов робот должен быть способен распознавать эти объекты как не принадлежащие ни к одному известному классу и запускать процесс адаптации к ним. При наличии манипулятора и видеокамеры, робот может осуществить это по алгоритму, структурная схема которого приведена на рис. 6.

Как только появляется объект, вектор которого лежит дальше установленного порога  $d_p$ , запускается процесс проверки: определяются координаты центра объекта в 3D-пространстве и выполняются  $N$  снимков этого объекта с разных ракурсов. Если среди векторов собранных изображений среднее минимальных расстояний  $\bar{d}_{i\min}$  меньше  $d_p$ , то объекту присваивается наиболее частый класс и собранные векторы добавляются в обучающую выборку. В противном случае запускается процесс дообучения: производится  $R$  снимков объекта, определяются их векторы признаков, которые затем добавляются в базу данных с меткой класса  $j+1$ , где  $j$  — текущее количество обученных классов. Из эвристических соображений следует выбирать параметры  $N$  и  $R$  таким образом, чтобы  $N < R$ , так как для уточнения классификации необходимо, как правило, меньше данных, чем на дообучение.

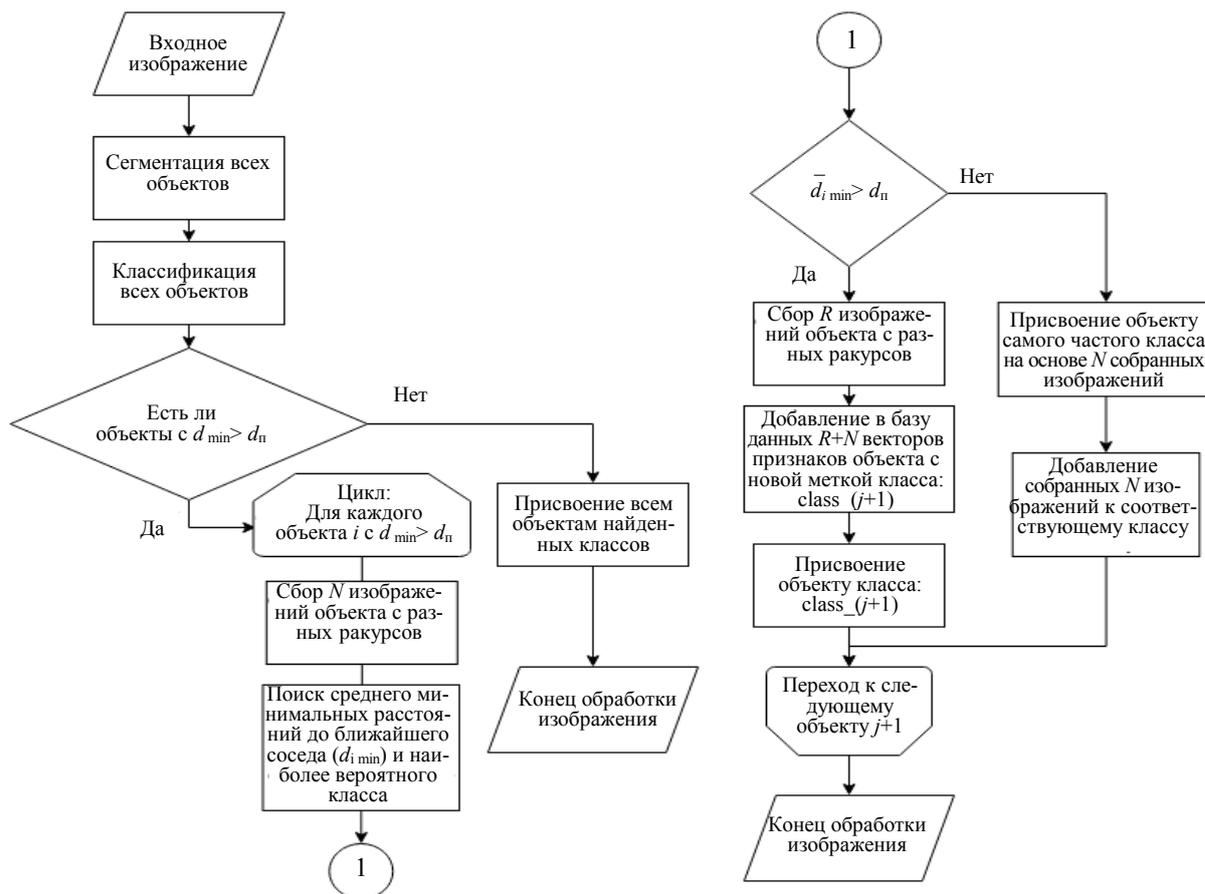


Рис. 6

**Заключение.** Исследована возможность применения предобученной модели для сегментации Mask R-CNN и обучения метрики для решения задачи расширения распознаваемых классов. В отличие от методики итеративного обучения данный метод не подвержен влиянию катастрофического забывания, а также не требует времени на дообучение.

Перспективные исследования связаны с соединением экстрактора признаков в Mask R-CNN и классификатора для ускорения работы алгоритма, а также с уменьшением количества сохраняемых точек в пространстве признаков и удалением выбросов в данных. Для улучшения качества сегментации планируется обучить Mask R-CNN на наборе данных, приспособленном для задач мобильной робототехники.

## СПИСОК ЛИТЕРАТУРЫ

1. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks // Advances in Neural Information Processing Systems / Ed.: F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger. Curran Associates Inc., 2012. Vol. 25. P. 1097—1105.
2. Deep Residual Learning for Image Recognition / Kaiming He, Xi-Anyu Zhang, Shaoqing Ren, Jian Sun // CoRR. 2015. Vol. ab-s/1512.03385. [Электронный ресурс]: <<http://arxiv.org/abs/1512.03385>>.
3. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications / A. G. Howard, Menglong Zhu, Bo Chen et al. // CoRR. 2017. Vol. abs/1704.04861. [Электронный ресурс]: <<http://arxiv.org/abs/1704.04861>>.
4. Mask R-CNN / Kaiming He, G. Gkioxari, P. Doll'ar, R. B. Girshick // CoRR. 2017. Vol. abs/1703.06870. [Электронный ресурс]: <<http://arxiv.org/abs/1703.06870>>.

5. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / *Shaoqing Ren, Kaiming He, R. Girshick, Jian Sun* // *Advances in Neural Information Processing Systems* / Ed.: *C. Cortes, N. Lawrence, D. Lee* et al. Curran Associates Inc., 2015. Vol. 28. P. 91—99. [Электронный ресурс]: <<https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.15>>.
6. *Kirkpatrick J., Pascanu R., Rabinowitz N.* et al. Overcoming catastrophic forgetting in neural networks // *Proc. of the National Academy of Sciences*. 2017. N 114(13). P. 3521—3526.
7. *Zenke F., Poole B., Ganguli S.* Continual Learning Through Synaptic Intelligence // *Proc. of the 34th Intern. Conf. on Machine Learning*. Sydney, Australia. 2017. Vol. 70. P. 3987—3995,
8. *Lomonaco V., Maltoni D.* CORE50: A New Dataset and Benchmark for Continuous Object Recognition // *Proc. of the 1st Annual Conf. on Robot Learning*. PMLR. 2017. Vol. 78. P. 17—26.
9. *Progressive Neural Networks* / *A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, R. Hadsell.* // arXiv preprint arXiv:1606.04671, 2016.
10. *Hayes T. L., Cahill N. D., Kanan C.* Memory Efficient Experience Replay for Streaming Learning // arXiv preprint arXiv:1809.05922, 2018.
11. *Rebuffi S. A., Kolesnikov A., Sperl G., Lampert C. H.* iCaRL: Incremental Classifier and Representation Learning // *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii. 2017.
12. *Hoffer E., Nir A.* Deep metric learning using Triplet network // *Intern. Workshop on Similarity-Based Pattern Recognition*. Springer, Cham, 2015.
13. *Columbia Object Image Library (COIL-100)* / *S. A. Nene, S. K. Nayar, H. Murase.* // *Tech. Report CUCS-006-96*. 1996. February.
14. *Wu Yuxin.* Detectron2, 2019. [Электронный ресурс]: <<https://github.com/facebookresearch/detectron2>>.
15. *Pytorch: An Imperative Style, High-Performance Deep Learning Library* / *A. Paszke, S. Gross, F. Massa* et al. // *Advances in Neural Information Processing Systems 32* / Ed.: *H. Wallach, H. Larochelle, A. Beygelzimer* et al. Curran Associates Inc., 2019. P. 8024—8035.
16. *Musgrave K. , Belongie S., Lim S.-N.* Pytorch metric learning // arXiv preprint, arXiv:2008.09164, 2020.

#### Сведения об авторах

- Иван Дмитриевич Ненахов** — студент; Университет ИТМО, факультет систем управления и робототехники, лаборатория биомехатроники и энергоэффективной робототехники; E-mail: ivdmne@gmail.com
- Кирилл Артемов** — аспирант; Университет ИТМО, факультет систем управления и робототехники, лаборатория биомехатроники и энергоэффективной робототехники; инженер-исследователь; E-mail: kaartemov@itmo.ru
- СейедХассан Забихифар** — канд. техн. наук; ПАО „Сбербанк“, лаборатория робототехники; инженер-разработчик; E-mail: zabikhifar.s@sberbank.ru
- Александр Николаевич Семочкин** — канд. физ.-мат. наук, доцент; ПАО „Сбербанк“, лаборатория робототехники; гл. инженер-разработчик; E-mail: Semochkin.A.N@sberbank.ru
- Сергей Алексеевич Колюбин** — д-р техн. наук, доцент; Университет ИТМО, факультет систем управления и робототехники, лаборатория биомехатроники и энергоэффективной робототехники; вед. научный сотрудник; E-mail: s.kolyubin@itmo.ru

Поступила в редакцию 28.12.21; одобрена после рецензирования 11.01.22; принята к публикации 18.01.22.

#### REFERENCES

1. Krizhevsky A., Sutskever I., Hinton G.E. *Advances in Neural Information Processing Systems*, 2012, vol. 25, pp. 1097—1105, <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.
2. He K., Zhang X., Ren Sh., Sun J. *CoRR*, 2015, vol. abs/1512.03385. <http://arxiv.org/abs/1512.03385>.
3. Howard A.G., Zhu M., Chen B. et al. *CoRR*, 2017, vol. abs/1704.04861, <http://arxiv.org/abs/1704.04861>.
4. He K., Gkioxari G., Dollár P., Girshick R.B. *CoRR*, 2017, vol. abs/1703.06870, <http://arxiv.org/abs/1703.06870>.
5. Ren Sh., He K., Girshick R., Sun J. *Advances in Neural Information Processing Systems*, 2015, vol. 28, pp. 91—99, <https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.15>.

6. Kirkpatrick J., Pascanu R., Rabinowitz N., Veness J., Desjardins G., Rusu A.A., Milan K., Quan J., Ramalho T., Grabska-Barwinska A., Hassabis D., Clopath C., Kumaran D., and Hadsell R. *Overcoming catastrophic forgetting in neural networks*, 2017, no. 13(114), pp. 3521–3526.
7. Zenke F., Poole B., and Ganguli S. *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, 2017, vol. 70, pp. 3987–3995.
8. Lomonaco V. and Maltoni D. *Proceedings of the 1st Annual Conference on Robot Learning*, PMLR, 2017, vol. 78, Proceedings of Machine Learning Research, pp. 17–26.
9. Rusu A.A., Rabinowitz N.C., Desjardins G., Soyer H., Kirkpatrick J., Kavukcuoglu K., Pascanu R., and Hadsell R. *arXiv preprint*, arXiv:1606.04671, 2016.
10. Hayes T.L., Cahill N.D., and Kanan Ch. *arXiv preprint*, arXiv:1809.05922, 2018.
11. Rebuffi S.-a., Kolesnikov A., Sperl G., and Lampert Ch.H. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, 2017.
12. Hoffer E. and Nir A. *International Workshop on Similarity-Based Pattern Recognition*, Springer, Cham, 2015.
13. Nene S.A., Nayar S.K., and Murase H. *Columbia Object Image Library (COIL-100)*, Technical Report CUCS-006-96, February 1996.
14. Wu Y. Detectron2, <https://github.com/facebookresearch/detectron2>, 2019.
15. Paszke A., Gross S., Massa F. et al. *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035.
16. Musgrave K., Belongie S., and Lim S.-N. *arXiv preprint*, arXiv:2008.09164, 2020.

**Data on authors**

<b>Ivan D. Nenakhov</b>	— Student; ITMO University, Faculty of Control Systems and Robotics, International Laboratory of Biomechatronics and Energy-Efficient Robotics; E-mail: ivdmne@gmail.com
<b>Kirill Artemov</b>	— Post-Graduate Student; ITMO University, Faculty of Control Systems and Robotics, International Laboratory of Biomechatronics and Energy-Efficient Robotics; Engineer-Researcher; E-mail: kaartemov@itmo.ru
<b>Seyedhassan Zabihifar</b>	— PhD; Sberbank, Robotics Laboratory; Engineer-Designer; E-mail: zabikhifar.s@sberbank.ru
<b>Aleksandr N. Semochkin</b>	— PhD, Associate Professor; Sberbank, Robotics Laboratory; Chief Engineer-Designer; E-mail: Semochkin.A.N@sberbank.ru
<b>Sergey A. Kolyubin</b>	— Dr. Sci., Associate Professor; ITMO University, Faculty of Control Systems and Robotics, International Laboratory of Biomechatronics and Energy-Efficient Robotics; Leading Researcher; E-mail: s.kolyubin@itmo.ru

Received 28.12.21; approved after reviewing 11.01.22; accepted for publication 18.01.22.