

<u>ИПМ им.М.В.Келдыша РАН</u> • <u>Электронная библиотека</u> <u>Препринты ИПМ</u> • <u>Препринт № 51 за 2020 г.</u>



ISSN 2071-2898 (Print) ISSN 2071-2901 (Online)

#### <u>Подопросветов А.В., Басс Л.П.,</u> Пластинин Ю.А., Скрябышева И.Ю.

Поиск и распознавание спутников на околоземных орбитах

*Рекомендуемая форма библиографической ссылки:* Поиск и распознавание спутников на околоземных орбитах / А.В.Подопросветов [и др.] // Препринты ИПМ им. М.В.Келдыша. 2020. № 51. 12 с. <u>http://doi.org/10.20948/prepr-2020-51</u> URL: <u>http://library.keldysh.ru/preprint.asp?id=2020-51</u>

## Ордена Ленина ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ имени М.В.Келдыша Российской академии наук

# А.В. Подопросветов, Л.П. Басс, Ю.А. Пластинин, И.Ю. Скрябышева

# Поиск и распознавание спутников на околоземных орбитах

#### УДК 004.93

#### Подопросветов А.В., Басс Л.П., Пластинин Ю.А., Скрябышева И.Ю. Поиск и распознавание спутников на околоземных орбитах

Исследование посвящено решению задачи обнаружения космических аппаратов на изображении для анализа их состояния путем визуального контроля. Предполагается выполнение задачи обслуживания одного космического аппарата на борту другого космического аппарата, в которую входит задача проверка их целостности. В работе представлен вариант приложения для поиска и распознавания спутников на изображениях космического пространства с помощью нейронных сетей. Для реализации программного комплекса используются открытые источники: обработанные и подготовленные фотографии спутников, находящихся на околоземных орбитах, инструмент для маркировки - присвоения имени объекту на фотографии, а также программная система *Detectron*, реализующая современные алгоритмы обнаружения объектов. В работе представлено описание собранной архитектуры и подготовленного набора данных для обучения и тестирования нейронной сети. Также приведены результаты исследования данной архитектуры и возможные варианты её доработки для решения поставленной задачи.

*Ключевые слова:* Анализ космических аппаратов, распознавание образов, нейронные сети, поиск космических аппаратов.

#### Podoprosvetov A.V., Bass L.P., Plastinin Yu.A., Skryabysheva I.Yu. Search and recognition of satellites in near-earth orbits

The study is dedicated to the problem of detecting and identifying spacecraft in the image for analyzing their state by visual inspection. The paper presents solution for the search and recognition of satellites in images of outer space using neural networks. To implement the software package, open sources are used: processed and prepared photographs of satellites in near-Earth orbits, a marking tool - naming an object in a photograph, as well as the Detectron software system that implements modern object detection algorithms. The paper presents a description of the collected architecture and the prepared data set for training and testing the neural network. The results of testing different architectures are presented and possible options for its refinement to solve the intare problem.

*Key words:* Spacecraft analysis, pattern recognition, neural networks, spacecraft search.

#### Введение

Современные космические исследования показывают, что на околоземных орбитах скапливается существенное количество космического мусора, который образуется из отработанных фрагментов ракетно-космической техники [1].

Одной из важнейших задач в данной области является наблюдение и мониторинг космического мусора. Существует множество подходов к его поиску [2] и далее к определению его движения [3] и очищению околоземного пространства. На протяжении последних десятилетий ведутся исследования по разработке алгоритмов маневрирования спутников на засоренных высотах [4] для предотвращения их столкновения с мусором.

В связи с целью решения задач обслуживания космических аппаратов в космосе ведется анализ состояния космического аппарата при его визуальном контроле методами технического зрения, а также определение оптимальных параметров перспективных систем технического зрения для решения целевых задач. Предполагается оснащать спутники такими системами, поэтому далее будет учтено тестирование выбранного решения на виртуальной машине аппаратной системы спутника, и на самом спутнике.

Однако на данном этапе можно рассматривать задачу поиска спутника на кадре для разных целей. Подробно в данной работе рассматривается наиболее сложная задача по определению точных контуров и точной идентификации спутника по изображению. Однако до описания этой задачи приведен краткий перечень более простых задач поиска спутников на изображении и подходы к их решению.

Авторами рассмотрена задача определения наличия спутников на изображении или его части и подход к решению, основанный на методе классификации изображений. В качестве тестового решения была применена нейронная сеть с двумя сверточными слоями [14]. На вход программе подавались разрезанные на четыре части изображения из открытого источника [5], подробное описание выборки использовавшихся изображений представлено ниже, на выходе требовалось определить, на какой именно части находится спутник. Обучение проводилось на четырех типах предобработанных данных, а именно, с предварительной фильтрацией и бинаризацией изображений алгоритмом Брэдли, без них, только с фильтрацией и только с бинаризацией изображений. Обучение проходило на выборке, состоящей как из изображений со спутником на фоне космического пространства и звезд, так и на фоне Земли. Удалось получить результат определения изображения со спутниками по метрике точности в 91% (ассигасу - близость измеренного значения к истинному значению или отношение истинно положительных и истинно отрицательных исходов ко всем исходам см. п4). Получить более высокую точность можно при введении новой классификации изображений, учитывающей изображения с небольшими частями спутника или частями антенны спутника.

Рассмотренна задача определения ориентации спутника на изображении относительно ориентации камеры в абсолютном пространстве. Для ее решения была реализована архитектура нейронной сети, основанная на подходе описанном в статье [12]. Однако на данном этапе исследования было реализовано решение для более грубой ее интерпретация в виде ответа на вопрос, какие грани спутника находятся в кадре. Данные, полученные при обучении на наборе данных из источника [5], показали результат по метрике "точность" в 94%.

Также отдельно была рассмотрена задача определения габаритов спутника в двух интерпретациях: классификация по площади занимаемой спутником на изображении и поиск рамки, заключающей в себе спутник. Архитектуры и алгоритмы нейронных сетей были построены на основе статьи [13]. Достигнутые точности для первой и второй интерпретации задачи составили соответственно 88% и 96%.

## 1. Постановка задачи

Целью работы является автоматизирование идентификации спутников и их неисправностей путем их визуального контроля на околоземной орбите. Предполагается выполнение задачи на борту космического аппарата. Задача обслуживания космических аппаратов включает проверку работоспособности космических аппаратов, проверку целостности спутника. Прежде всего рассматривается задача поиска искусственного спутника на монохромном изображении. Выборка состоит из изображений спутников на фоне космического пространства или на фоне Земли. Под космическим пространством будем понимать пространство со звездами и Землей. Результатом поиска искусственного спутника является определение точных контуров спутника, а именно массив 2d координат контуров на изображении.

Для этих целей требуется построить или использовать известную архитектуру нейронной сети, входными данными которой является монохромное изображение, а на выходе имеется точный контур — маска, которая покрывает весь спутник на изображении при его наличии, а также рамка, которая обводит область спутника с указанием его типа. Специальным маркером необходимо определять отсутствие спутника на изображении. Также необходимо заложить в модель возможное присутствие различных спутников и оценку вероятности их определения. Стоит отметить, что задача определение вида спутника не рассматривалась, поскольку для ее решения необходимы параметры разных спутников или обучающая выборка с их примерами.

## 2. Обучающая выборка

За основу для обучающей и проверочной выборки изображений был взят открытый для общего доступа ресурс [5]. Данный ресурс содержит пятнадцать тысяч синтетических монохромных изображений в формате *jpeg* с разрешением 1920х1200 пикселей, из которых двенадцать тысяч имеют информацию об ориентаций спутника на кадре. Для решения рассматриваемой задачи из имеющегося ресурса была подготовлена обучающая выборка, состоящая из одной тысячи изображений. Стоит заметить, что на всех изображениях присутствует один и тот же спутник. Выборка была получена при помощи инструмента *labelme* [6], используемого для аннотирования — подписи названия или семантического значения графических объектов на изображении. Данный инструмент написан на языке программирования Python с использованием библиотек PyQt. В инструменте на объект, расположенный на изображении, в качестве лейбла по его контуру накладывается полупрозрачная маска и ключевая рамка с названием. Там же предусмотрен лейбл — "задний фон", который также является объектом изображения.



Рис. 2.1: Примеры из обучающего набора данных

Заметим, что при дальнейших исследованиях может возникнуть необходимость наложения фильтров против солнечной засветки, которое не требовалось на данном этапе.

Для сбора изображений с лейблами было написано приложение для упрощения процедуры разметки в инструменте *lableme*. Данное приложение создает набор контуров в формате *.json* с помощью применения маски, созданной с использованием оператора Кэнни (определение границ объекта) из библиотеки *opencv* [10].

Сбор изображений в обучающую выборку производится в два этапа. На первом этапе каждое изображение обрабатывается написанным приложение ем с подстройкой параметров для получения наиболее четких контуров. На втором этапе изображения и контуры загружались в приложение *labelme* и редактировались вручную. Тем самым, был получен набор данных — обучающая выборка формата *COCO*.

Примеры изображений из обучающей выборки приведены на рисунке (2.1.)

#### 3. Программный комплекс

Программный комплекс реализован в облачном сервисе на языке программирования Python с использованием библиотек: pytorch (библиотека машинного обучения для языка Python с открытым исходным кодом, созданная на базе Torch), tensor flow (открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия), cocoAPI, cython и detectron2. Основным модулем является detectron2 (программная система, разработанная командой FacebookAIResearch, в которой реализованы самые современные алгоритмы обнаружения объектов), остальные модули — вспомогательные. В качестве облачного сервиса (удобного использования мощных удаленных вычислительных ресурсов) используется GoogleColab с использованием графического процессора gpu Nvidia Tesla K80, предоставленного сервисом. Использование вычислительной мощности облачного сервиса, позволяет значительно ускорить процесс обучения нейросети. К сожалению, перенос архитектуры для обучения из облачного сервиса, является сложной задачей, связанным с использованием библиотек CUDA. Для обеспечения быстрой и более удобной работы облачного сервиса была организована загрузка изображений для обучения и проверки точности нейронной сети из GoogleDrive.

В модуле detectron2 представлена высококачественная, высокопроизводительная кодовая база для исследований, направленных на обнаружение объектов. В модуле собраны основные архитектуры на базе сверточных нейронных сетей (CNN). Помимо архитектур модуль располагает основными алгоритмами для распознавания объектов. Одним из наилучших алгоритмов является MaskR - CNN [7], который расширяет широко используемый метод FasterR - CNN [8], добавляя ветвь для предсказания маски объекта параллельно с существующей ветвью для распознавания ограничивающего прямоугольника. Данный алгоритм использует архитектуру сети ResNet + FPN со стандартными заголовками conv и FC для предсказания маски и блока, соответственно. MaskR - CNN прост в обучении и добавляет незначительные расходы ресурсов к методу FasterR - CNN, работающему со скоростью 5 кадров в секунду. Кроме того, метод MaskR - CNN легко обобщается для других задач классификации, что позволит использовать его в будущих исследованиях. В данном исследовании было проведено сравнение трех архитектур: FasterR - CNN с R101 - FPN, RetinaNet с R101 и MaskR - CNNс R101 - FPN. Результаты сравнения моделей на поставленной задаче представлены после описания метрик их сравнения.

## 4. Описание метрик эффективности архитектур

Для сравнения выбранных моделей используются метрика точности mAP выбранного надора данных в формате *COCO*. Метрика содержит три этапа вычислений. Для первого этапа используются определения. Оценка достоверности — это вероятность того, что ключевая рамка содержит объект. Вероятность предсказывается методом машинного обучения (MO).

Пересечение через объединение — это мера, основанная на индексе Жакара (IoU), определяется как отношение площади пересечения к площади объединения, ограниченных предсказанной ключевой рамкой ( $B_p$ ) и заданной рамкой ( $B_{gt}$ ), как и в теории множеств, объединение не учитывает область пересечения геометрических фигур дважды:

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{qt})}.$$
(4.1)

Данные определения используются в качестве критериев, определяющих, является ли обнаружение истинно положительным (ИП - *TP*) или ложно положительным (ЛП - *FP*). Обнаружение считается ИП только при условии выполнения трёх критериев:

- показатель достоверности больше порога;
- предсказанный класс соответствует истинному классу объекта;
- прогнозируемая ограничивающая рамка имеет *IoU*, превышающий пороговое значение (например, 0,5) с заданной рамкой.

Обнаружение считается  $\Pi (FP)$ , если третье условие не выполняется, а именно прогнозируемая ограничивающая рамка имеет IoU, не превышающий пороговое значение.

Для второго этапа считаются два вспомогательных параметра. Первый из них точность (*Precision*), он определяется, как количество истинно положительных результатов по отношению к сумме истинных положительных и ложных положительных результатов.

$$Precision = \frac{\Pi\Pi}{\Pi\Pi + \Pi\Pi}.$$
(4.2)

Второй — полнота (*Recall*), или чувствительность, он определяется как число истинно положительных результатов по отношению к сумме истинно положительных и ложно отрицательных значений (ЛО - FN). Следует обратить внимание, что сумма — это просто число истинностей, поэтому нет необходимости подсчитывать количество ложно отрицательных значений.

$$Recall = \frac{\Pi\Pi}{\Pi\Pi + \PiO}.$$
(4.3)

В свою очередь, кривая зависимости точности от полноты может быть использована для оценки методов МО, но сложно сравнивать разные методы МО когда кривые по ним пересекаются. Поэтому принято рассматривать среднюю точность (СТ - AP), основанную на данной кривой. Другими словами, AP — это точность, усредненная по всем уникальным уровням полноты. Для уменьшения колебаний кривой сначала рассматривают интерполяцию точности по полноте, а затем считают среднюю точность AP, при условии, что в качестве интерполяции берется максимум кривой на каждом уровне полноты.

$$p_{interpolated} = \max_{r' \ge r} p(r). \tag{4.4}$$

Далее средняя точность определяется, как область под интерполированной кривой точности-полноты, которая может быть рассчитана по следующей формуле:

$$AP = \sum_{i=1}^{n-1} (r_{i+1} + r_i) p_{interpolated}(r_{i+1}), \qquad (4.5)$$

где  $r_i$  — это уровни полноты в порядке возрастания, которыми интерполируется точность. В классификаторах применяется больше одного класса например k классов. Средняя точность (*AP*) считается по каждому классу в отдельности. Принято рассматривать в качестве третьего этапа усредненную по классам среднюю точность mAP

$$mAP = \frac{\sum_{i=1}^{k} AP_i}{k}.$$
(4.6)

В данном исследовании было проведено сравнение трех архитектур: Faster  $R - CNN \ R101 - FPN$ , RetinaNet R101 и Mask $R - CNN \ R101 - FPN$ . Их сравнительная характеристика приведена в таблице 1.

Сравнение современных сверточных нейросетей для методов Object Detection на наборе в формате СОСО. Таблица 1.

	AP	AP50	AP75	APs	APm	APL
Faster R-CNN R101-FPN	35.2	57.3	38.0	19.2	39.3	47.1
RetinaNet R101	37.1	60.1	42.2	21.9	9 41.3	4.3
Mask R-CNN R101-FPN	41.8	62.3	<b>42.2</b>	25.1	42.7	48.2

Были проведены измерения при различных коэффициентах Жаккара (AP50 - при коэффициенте 50%, AP75 - при коэффициенте 75%. APs - для объектов площадью менее 32 квадратных пикселя, <math>APm - для объектов площадью более 32, но менее 96 квадратных пикселя, APL - для объектов большей площади). Из результатов исследования видно, что максимальная точность достигнута моделью MaskR - CNN.

#### 5. Полученные результаты

В работе рассмотрена задача обнаружения космических аппаратов на изображениях, определения их вида и типа и проверки их целостности. В работе приведены результаты испытания простых архитектур нейронных сетей для некоторых подзадач описанных во введении. Основной частью работы является собранная и обработанная обучающая выборка, основанный на открытом источнике [5], а также сравнительный анализ и тестирование архитектур в применении к описанной задаче. В работе приведены метрики тестирования на выбранных архитектурах, и описана методика их получения. Таким образом к новой задаче был успешно применен известный метод. Ясно, что используемый инструмент требует доработок в связи с установкой данной системы на борту космического аппарата, а также в связи с адаптацией данной «стендовой» работы к практическому применению в условиях ограниченных аппаратных средств и энергозатрат.

По результатам обучения на семистах изображениях классификатором было распознано три класса: земля, спутник и задний фон в виде космоса. Предполагается, что в дальнейшей работе спутники будут разных видов и типов. На приведенных примерах работы алгоритма, представленных на рисунке (5.1), изображены спутники, занимающие качественно разную площадь изображения, выделена маска их контура, показана ключевая рамка с подписью названия спутника и вероятностью определения его типа в левом верхнем углу рамки (мелко), увеличенный прямоугольник на первом изображении.



Рис. 5.1: Примеры работы нейронной сети

Поскольку на данном этапе для тестирования алгоритма использовался один тип спутника, то вероятность определения его типа составила 99-100%.

## Список литературы

- 1. Шустов Б.М. О фундаментальных исследованиях по проблеме космического мусора // Всероссийская научная конференция «Космический мусор: фундаментальные и практические аспекты угрозы»: сборник трудов. Серия «Механика, управление и информатика». М.: ИКИ РАН, 2019. С.7-14. doi:10.21046/spacedebris2019-7-14
- 2. Space debris detection in optical image sequences / Jiangbo Xi at al. // Appl. Opt. 55, 7929-7940 (2016) https://doi.org/10.1364/AO.55.007929
- Resident Space Object Characterization and Behavior Understanding 3. Machine Learning and Ontology-based Bayesian via Networks / Roberto Furfaro et al. // Conference: Advanced Maui Optical and Space Surveillance Technologies ConferenceAt: Maui. HI. 2016. https://www.researchgate.net/publication/309491642 (дата обращения: 01.07.2020).
- 4. Зверев П.С., Довгаль В.М. Метод и алгоритм распознавания искусственных околоземных орбитальных объектов и «Mycopa» ДЛЯ безопасности космических полетов Вестник ВΓобеспечения https://cyberleninka.ru/article/n/metod-i-algoritm-TУ. 2010. <u>№</u>4. raspoznavaniya-iskusstvennyh-okolozemnyh-orbitalnyh-obektov-i-musoradlya-obespecheniya-bezopasnosti-kosmicheskih обращения: (дата 01.07.2020).
- 5. https://kelvins.esa.int/satellite-pose-estimation-challenge/data/
- 6. *Kentaro Wada* labelme: Image Polygonal Annotation with Python (2016) https://github.com/wkentaro/labelme
- 7. Kaiming He, Georgia Gkioxari, Piotr Dollár and Ross Girshick Mask R-CNN, 2017, https://arxiv.org/abs/1703.06870
- 8. Ross Girshick Fast R-CNN, 2015, https://arxiv.org/abs/1504.08083
- H. Zeng and Y. Xia , Space target recognition based on deep learning // 2017 20th International Conference on Information Fusion, Xi'an, 2017, pp. 1-5, doi: 10.23919/ICIF.2017.8009786
- 10.  $https://docs.opencv.org/master/d6/d00/tutorial_py_root.html$
- 11. Yuxin Wu, Alexander Kirillov, Francisco Massa, et al., Detectron2, 2019, https://github.com/facebookresearch/detectron2
- 12. Shih-En Wei, Varun Ramakrishna, Takeo Kanade and Yaser Sheikh, Convolutional Pose Machines, 2016, https://arxiv.org/abs/1602.00134

- Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan and Serge Belongie Feature Pyramid Networks for Object Detection, 2016, https://arxiv.org/abs/1602.00134
- 14. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. https://www.tensorflow.org/api docs/python/tf/keras/layers/Conv2D

## Содержание

Введ	цение	3
1.	Постановка задачи	4
2.	Обучающая выборка	5
3.	Программный комплекс	6
4.	Описание метрик эффективности архитектур	7
5.	Полученные результаты	9
Спис	сок литературы	1