

А. В. Смирнов, Е. С. Иванов

Использование механизма сверточных нейронных сетей для поиска объектов на аэрофотоснимках

Аннотация. В статье описан метод поиска объектов на аэрофотоснимках с применением нейронных сетей, а также алгоритм, позволяющий выполнить постобработку данных, полученных в результате работы нейронных сетей. Рассмотрена задача поиска самолетов на изображениях.

Ключевые слова и фразы: нейронные сети, поиск объектов, аэрофотоснимки.

Введение

Задача нахождения объектов интереса на изображениях до сих пор актуальна, особенно в области мониторинга и анализа снимков дистанционного зондирования земли (ДЗЗ) или аэрофотоснимков. Для решения данной задачи используются различные алгоритмы от простой пороговой бинаризации [1], [2] до применения гистограмм направленных градиентов [3] и методов Виолы-Джонса [4].

В последнее время большое распространение получили нейронные сети [5], [6] теория о которых появилась ещё во второй половине XX века. Различные архитектуры нейронных сетей используются не только для нахождения объектов интереса, но и для классификации, распознавания объектов, поиска нескольких типов интересующих объектов.

Работа выполнена в рамках Договора № 124-ВС06/13/423 «Исследования и разработка архитектурных, аппаратных и программных технических решений и создание экспериментальных средств блочно-параллельной обработки космической информации в высокопроизводительных вычислительных системах с производительностью кластера не менее 400 Гфлопс» (шифр: Мониторинг-СГ-2.5.3.1).

© А. В. Смирнов, Е. С. Иванов, 2017

© Институт программных систем имени А. К. Айламазяна РАН, 2017

© Программные системы: теория и приложения, 2017

DOI: 10.25209/2079-3316-2017-8-4-85-99

В данной статье будет рассмотрен пример использования нейронной сети для нахождения различных видов самолётов, находящихся на территории аэропорта. Полученная в результате информация после постобработки может использоваться в задачах мониторинга взлётно-посадочных полос, например, для отслеживания перемещения самолётов или подсчёта их количества.

Обработка информации происходила в системе блочно-параллельной обработки данных [7]. Система имеет модульную архитектуру. Под модулем подразумевается программный компонент, который обрабатывает данные, поступающие по входному каналу и передает обработанную информацию в выходной канал. Задача системы — обеспечить обработку данных в конвейерно-параллельном режиме.

Этот подход ускоряет обработку данных с использованием суперкомпьютера. На вход система получает описание задачи, которое содержит список модулей обработки и параметры запуска. Система автоматически определяет, какие модули запускать и обеспечивает обмен данных между модулями.

1. Поиск объектов с помощью нейронных сетей

Использование нейронных сетей обусловлено с их схожестью с успешной работой биологических систем, которые, по сравнению с другими системами, состоят из простых и многочисленных нейронов, которые работают параллельно и имеют возможность обучения [8]. Для работы была выбрана сверточная нейронная сеть — специальная архитектура искусственных нейронных сетей, предложенная Яном Лекуном и нацеленная на эффективное распознавание изображений. Сверточная нейронная сеть состоит из пар слоев — слоев подвыборки и слоев свертки, каждый из которых в свою очередь состоит из карт признаков. Каждая карта признаков в идеале фильтрует изображение, находя какой-то один определенный, специфичный для данной карты, признак ([9]). Количество слоев в используемой нейронной сети составило 13. Обучение происходило на выборке, состоящей из четырех классов: дороги, растительность, городские застройки и самолёты. Общее количество образцов изображений для обучающей

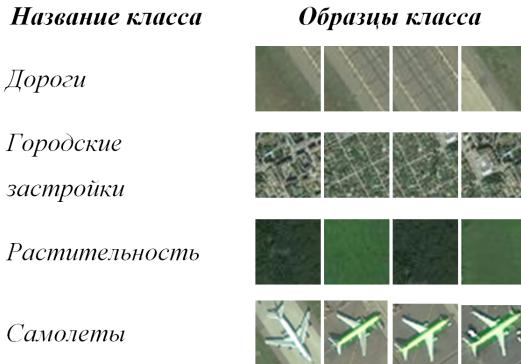


Рис. 1. Образцы классов для обучения нейронной сети

выборки составило около 5000. Размер каждого изображения (при необходимости) преобразовывался в 32×32 .

На рис. 1 представлены образцы обучающей выборки.

Количество эпох обучения составило 5000. Ошибка обучения составила 0,2%.

Схема, описывающая решение задачи обучения нейронной сети в системе блочно-параллельной обработки данных показана на рис. 2 и состоит из последовательного выполнения модулей:

ReadMultiImage_array. Модуль, который считывает множество эталонов из указанных директорий и запоминает их принадлежность к тому или иному классу, согласно считанным данным.

FindObject_CNN. модуль, содержащий в себе архитектуру нейронной сети с возможностью изменения гиперпараметров сети, таких как количество слоёв свёртки, количество классов и т.д. Также модуль имеет параметр, значение которого определяет будет ли сеть обучаться (дообучаться) или перейдёт в режим поиска объектов интереса. Дообучение позволяет обучить нейронную сеть на небольшом объёме данных с сохранением состояния сети при завершении предыдущего обучения. В данной задаче параметр задан на обучение.

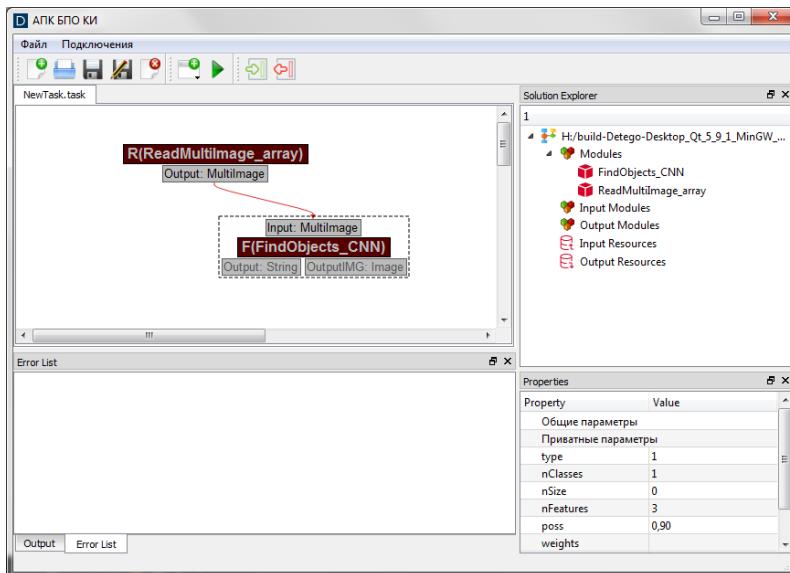


Рис. 2. Схема, решающая задачу обучения нейронной сети

Множество подходов, связанных с поиском объекта заданного класса на изображениях при помощи нейронных сетей, состоят из следующих этапов: выделение объектов (чаще всего используются гистограммы SIFT, HOG), после чего выделенный объект подается нейронной сети на распознавание [10], [11]. Недостатком этого подхода является то, что не всегда объекты могут выделяться при помощи таких гистограмм, как следствие — не все объекты могут быть найдены на изображении (возможна ошибка как в выделении объектов, так и в распознавании).

Для того, чтобы избежать ошибку с выделением объектов при помощи гистограмм, был предложен следующий алгоритм поиска объектов. Изображение обходится скользящими масками заданных размеров $N \times N$ с шагом $N/2$. Каждый из фрагментов изображения под скользящей маской подается нейронной сети для распознавания. Если вероятность отнесения рассматриваемого фрагмента к некому классу K больше, чем заданное пороговое значение p_0 , то считается, что на заданном участке изображен объект класса K . После того, как

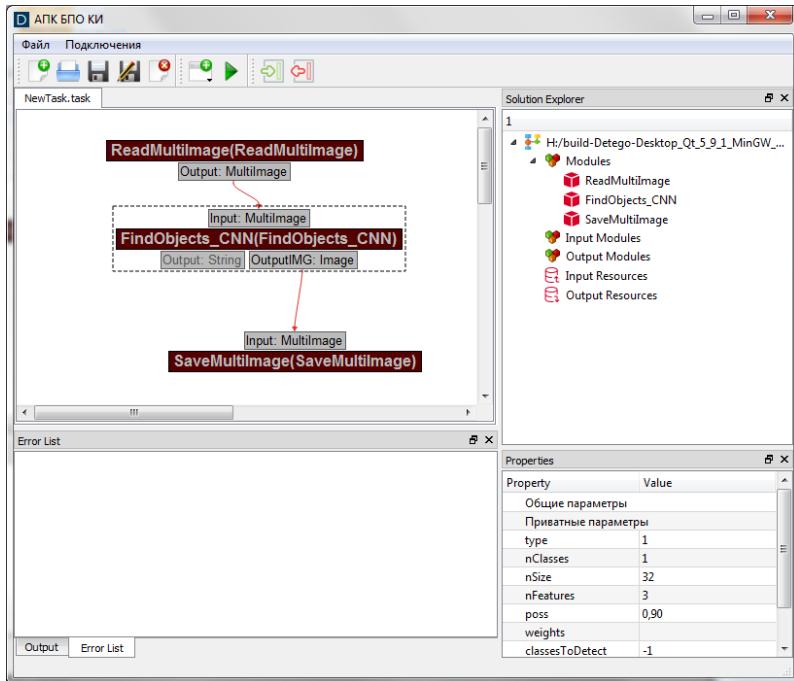


Рис. 3. Схема, решающая задачу поиска объектов с помощью нейронной сети

все фрагменты отнесены к какому-либо классу, происходит фильтрация по заданному пользователем классу: отбрасываются те участки изображения, на которых не изображены объекты искомого класса. Схема поиска с помощью нейронной сети состоит из последовательного выполнения модулей:

ReadMultiImage. Модуль чтения изображения.

FindObjects_CNN. модуль сверточной нейронной сети, запускаемой на обнаружение.

SaveMultiImage. Модуль сохранения результатов, найденных при помощи нейронной сети.

Схема в графическом интерфейсе представлена на рис. 3.



Рис. 4. Примеры результата поиска самолетов с помощью скользящих масок и нейронной сети

Задача поиска объектов разных масштабов является достаточно трудной в настоящее время [5]. На рис. 4 показаны результаты работы поиска объектов класса «самолет» с размерами скользящих масок 150×150 , 100×100 , 80×80 , 64×64 , 32×32 .

В связи с тем, что обход изображения происходит с перекрытием, повышается способность поиска объекта. Но при этом появляется возможность многократного вхождения одного и того же объекта сразу в несколько фрагментов. Такой результат работы нейронной сети тяжёл для восприятия и обработки, поэтому был предложен алгоритм слияния прямоугольников, содержащих объекты.

2. Алгоритм слияния координатных прямоугольников с учётом вероятности распознавания

Под координатным прямоугольником будем считать прямоугольник, наложенный на найденный объект, заданный шириной и высотой, являющимися габаритами искомого объекта, а также двумерными

(x, y) координатами левого верхнего угла прямоугольника. Предполагается, что координаты точки центра такого прямоугольника совпадают, либо почти совпадают в допустимой погрешности, с координатами точки центра найденного объекта.

Помимо координатных прямоугольников, нейронная сеть, после обработки изображения на выходе даёт вероятность, которая сопоставляется каждому из прямоугольников. Этой вероятностью определяется возможность нахождения координатного прямоугольника над искомым объектом, она представляет собой число с плавающей точкой и принимает значения от 0 до 1. Представленный алгоритм принимает на вход список координатных прямоугольников, а также сопоставленные им вероятности. Алгоритм слияния состоит из двух этапов:

- (1) Создание подсписков из основного списка координатных прямоугольников. Из основного списка выбирается опорный прямоугольник с нулевым индексом (опорный прямоугольник). Далее высчитываются координаты центра опорного прямоугольника. После этого рассчитывается Евклидово расстояние между центром опорного прямоугольника и центрами оставшихся координатных прямоугольников. Все прямоугольники, у которых расстояния от координат точки собственного центра до центра опорного прямоугольника ниже некого порога (порог подбирается опытным путём), добавляются в новый список с опорным прямоугольником, а из основного удаляются. Это происходит до тех пор, пока в основном списке не останется элементов.
- (2) Фильтрация координатных прямоугольников по вероятности и создание общего координатного прямоугольника. Каждый подсписок, полученный на предыдущем этапе, фильтруется по определённому порогу вероятности. Следовательно, в списке остаются координатные прямоугольники с высокой вероятностью. Затем из них формируется общий координатный прямоугольник, точка центра которого является средней арифметической точкой отфильтрованных прямоугольников. Габариты (ширина и высота) общего координатного прямоугольника можно вычислить двумя способами:

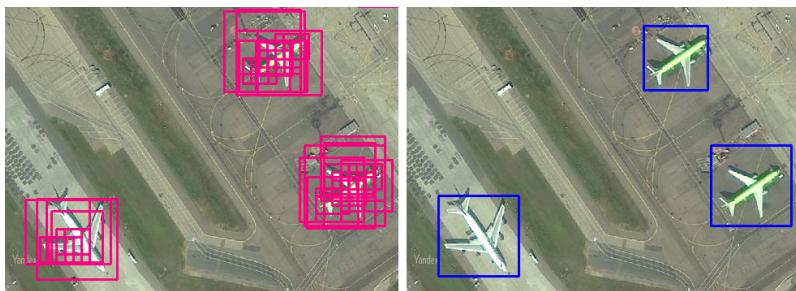


Рис. 5. Выделенные объекты интереса до (слева) и после (справа) применения алгоритма слияния

- среднее арифметическое габаритов отфильтрованных прямоугольников;
- габариты самого большого координатного прямоугольника из отфильтрованных.

На рис. 5 слева показано изображение с нанесёнными координатными прямоугольниками до применения алгоритма слияния, а справа после. В данном примере объектом интереса являются самолёты на взлётно-посадочной полосе.

Несмотря на то, что алгоритм состоит всего из двух этапов, не обязательно дожидаться полного выполнения первого этапа. Допускается запуск второго этапа сразу после получения первого подсписка. В этом случае можно сразу получить общий координатный прямоугольник, но второй этап потребуется запустить для каждого нового подсписка.

В проведенных экспериментах были случаи, когда подобная обработка снижала количество выделенных прямоугольников с 167 до 12.

3. Адаптивное изменение порога вхождения

Ранее в статье был указан порог вхождения (T), который подбирался опытным путём. Пороговым значением определялось, входит ли координатный прямоугольник в группу своих соседей или нет. Однако

ТАБЛИЦА 1. Сравнение значения опытного порога с рассчитанными значениями

T (подобранный в опытах)	T1 (СА)	T2 (СГ)
150	160	160
150	155	154.172
75	71.667	63.6043
80	79.9231	76.3799
45	43.0112	40.0211

вручную задавать порог для каждого нового изображения нецелесообразно. Чтобы избежать данной проблемы был разработан метод адаптивного изменения порога, основанный на анализе размеров координатных прямоугольников и сопоставляемой им вероятности. По сути, размеры координатных прямоугольников являются размерами квадратного сканирующего окна (маски) нейронной сети. Нейронная сеть делает несколько проходов по входному изображению используя маски разных размеров. Это делается для того, чтобы объекты интереса (в данном случае самолёты) разных размеров также успешно находились.

В ходе экспериментов с ручным подбором порога выяснилось, что значение порога зависит от размеров (можно использовать только ширину маски, так как она квадратная) используемых масок. Поэтому, метод адаптивного изменения порога заключается в отборе масок с наиболее высокой вероятностью и последующим их анализом. В качестве анализа используется:

- подсчёт среднего арифметического (СА) по ширине отобранных масок,
- подсчёт среднего геометрического (СГ) по ширине отобранных масок.

В таблице 1 показано сравнение значения подобранныго порога вручную с полученным адаптивным порогом.

Общая схема поиска объектов на изображении с применением фильтрации координатных прямоугольников представлена на рис. 6 и состоит из последовательного выполнения модулей:

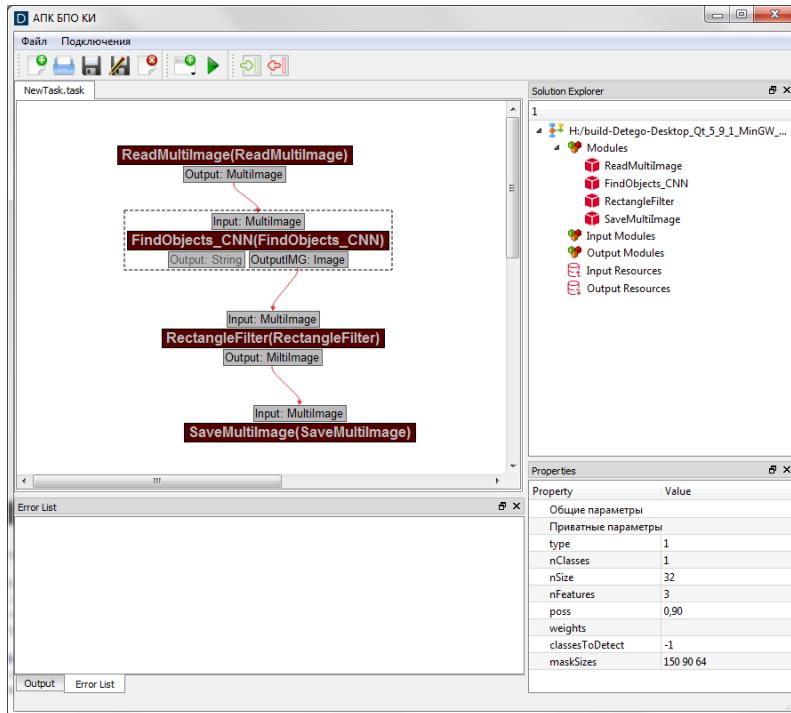


Рис. 6. Схема, решающая задачу поиска объектов с помощью нейронной сети с применением алгоритма фильтрации координатных прямоугольников

ReadMultiImage. Модуль чтения изображения.

FindObjects_CNN. Модуль сверточной нейронной сети, запускаемой на обнаружение.

RectsngleFilter. Модуль, выполняющий фильтрацию координатных прямоугольников.

SaveMultiImage. Модуль сохранения результатов, найденных при помощи нейронной сети.

На рис. 7 приведены примеры работы алгоритма с применением адаптивного порога.



Рис. 7. Результат поиска объектов с помощью нейронной сети и алгоритма фильтрации координатных прямоугольников

Заключение

Рассмотрен подход к поиску объектов на аэрофотоснимках с помощью нейронных сетей, используя скользящие маски, на примере поиска самолетов. Предложен алгоритм, постобработки результатов поиска объектов на изображении. Данный подход позволяет объединить подмножество координатных прямоугольников соответствующего объекта в один. В отдельных случаях было достигнуто сокращение количества координатных прямоугольников в 14 раз. Предложенный подход был реализован системе блочно-параллельной обработки данных.

Список литературы

- [1] Qiang Chen, Quan-sen Sun, Pheng Ann Heng, De-shen Xia. “A double-threshold image binarization method based on edge detector”, *Pattern Recognition*, **41**:4 (2008), pp. 1254–1267. ↑⁸⁵
- [2] J. Pitkänen. “Individual tree detection in digital aerial images by combining locally adaptive binarization and local maxima methods”, *Canadian Journal of forest research*, **31**:5 (2001), pp. 832–844. ↑⁸⁵
- [3] A. Rosebrock. *Histogram of Oriented Gradients and Object Detection* (Retrieved August 31, 2015), URL: <http://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/> ↑⁸⁵
- [4] *Face Detection using Haar Cascades*, OpenCV. Open Source Computer Vision Documentation, URL: http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html ↑⁸⁵
- [5] G.E. Hinton, N. Srivastava et al. *Improving neural networks by preventing co-adaptation of feature detectors*, 2012, URL: <https://arxiv.org/abs/1207.0580> ↑^{85,90}
- [6] G. Cheng, P. Zhou, J. Han. “Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images”, *IEEE Transactions on Geoscience and Remote Sensing*, **54**:12 (2016), pp. 7405–7415. ↑⁸⁵
- [7] A. Kondratyev, I. Tishchenko. “Concept of Distributed Processing System of Images Flow”, *Robot Intelligence Technology and Applications 4*, Results from the 4th International Conference on Robot Intelligence Technology and Applications, RiTA 2015 (Bucheon, Korea, December 14–16, 2015), Advances in Intelligent Systems and Computing, vol. **447**, eds. J.-H. Kim, F. Karray, J. Jo, P. Sincak, H. Myung, 2016, pp. 551–561. ↑⁸⁶
- [8] D. Kriesel. *A Brief introduction to Neural Networks*, ZETA2-EN, 2007, URL: http://www.dkriesel.com/_media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf ↑⁸⁶
- [9] Я.Ю. Дорогий. «Архитектура обобщенных сверточных нейронных сетей», *Вестник НТУУ КПИИ*, 2012, №57, 6 с., URL: http://www.it-visnyk.kpi.ua/wp-content/uploads/2012/08/54_36.pdf ↑⁸⁶
- [10] S. Ren et al. “Object detection networks on convolutional feature maps”, *IEEE transactions on pattern analysis and machine intelligence*, **39**:7 (2017), pp. 1476–1481. ↑⁸⁸
- [11] R. B. Girshick, J. Donahue, T. Darrell, J. Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”, *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2014 (Columbus, Ohio, USA, June 23–28, 2014), 2014, pp. 580–587, URL: https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.pdf ↑⁸⁸

Рекомендовал к публикации

д.ф.-м.н. С.В. Знаменский

Пример ссылки на эту публикацию:

А. В. Смирнов, Е. С. Иванов. «Использование механизма сверточных нейронных сетей для поиска объектов на аэрофотоснимках», *Программные системы: теория и приложения*, 2017, 8:4(35), с. 85–99.

URL: http://psta.psiras.ru/read/psta2017_4_85-99.pdf

Об авторах:



Александр Владимирович Смирнов

Инженер-исследователь Лаборатории методов обработки и анализа изображений, Институт Программных Систем имени А. К. Айламазяна РАН

e-mail:

asmirnov_1991@mail.ru



Егор Сергеевич Иванов

Инженер-исследователь Лаборатории методов обработки и анализа изображений, Институт Программных Систем имени А. К. Айламазяна РАН

e-mail:

egor.s.ivanov@gmail.com

Alexandr Smirnov, Egor Ivanov. *Objects detection on aerial photos usin neural networks.*

ABSTRACT. The paper describes the method of searching for objects on aerial photographs using neural networks, as well as an algorithm that allows postprocessing of data obtained as a result of the operation of neural networks. The problem of searching for aircraft in images is considered. *(In Russian).*

Key words and phrases: object detection, neural networks, aerial photos.

References

- [1] Qiang Chen, Quan-sen Sun, Pheng Ann Heng, De-shen Xia. “A double-threshold image binarization method based on edge detector”, *Pattern Recognition*, **41**:4 (2008), pp. 1254–1267.
- [2] J. Pitkänen. “Individual tree detection in digital aerial images by combining locally adaptive binarization and local maxima methods”, *Canadian Journal of forest research*, **31**:5 (2001), pp. 832–844.
- [3] A. Rosebrock. *Histogram of Oriented Gradients and Object Detection* (Retrieved August 31, 2015), URL: <http://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/>
- [4] *Face Detection using Haar Cascades*, OpenCV. Open Source Computer Vision Documentation, URL: http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html
- [5] G. E. Hinton, N. Srivastava et al. *Improving neural networks by preventing co-adaptation of feature detectors*, 2012, URL: <https://arxiv.org/abs/1207.0580>
- [6] G. Cheng, P. Zhou, J. Han. “Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images”, *IEEE Transactions on Geoscience and Remote Sensing*, **54**:12 (2016), pp. 7405–7415.
- [7] A. Kondratyev, I. Tishchenko. “Concept of Distributed Processing System of Images Flow”, *Robot Intelligence Technology and Applications 4*, Results from the 4th International Conference on Robot Intelligence Technology and Applications, RiTA 2015 (Bucheon, Korea, December 14–16, 2015), Advances in Intelligent Systems and Computing, vol. **447**, eds. J.-H. Kim, F. Karray, J. Jo, P. Sincak, H. Myung, 2016, pp. 551–561.
- [8] D. Kriesel. *A Brief introduction to Neural Networks*, ZETA2-EN, 2007, URL: http://www.dkriesel.com/_media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf
- [9] Ya. Yu. Dorogiy. “Arkhitektura obobshchennykh svertochnykh neyronnykh setey”, *Vestnik NTUU KPII*, 2012, no.57, 6 p., URL: http://www.it-visnyk.kpi.ua/wp-content/uploads/2012/08/54_36.pdf

- [10] S. Ren et al. “Object detection networks on convolutional feature maps”, *IEEE transactions on pattern analysis and machine intelligence*, **39**:7 (2017), pp. 1476–1481.
- [11] R. B. Girshick, J. Donahue, T. Darrell, J. Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”, *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014* (Columbus, Ohio, USA, June 23–28, 2014), 2014, pp. 580–587, URL: https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.pdf

Sample citation of this publication:

Alexandr Smirnov, Egor Ivanov. “Objects detection on aerial photos usin neural networks”, *Program systems: Theory and applications*, 2017, **8**:4(35), pp. 85–99. (In Russian).

URL: http://psta.psiras.ru/read/psta2017_4_85-99.pdf