

. . , . .

**Введение.** Пожары (степные, лесные и торфяные) – опасные природные явления, в ходе которых повреждаются или гибнут лесные насаждения, страдают населённые пункты, погибают люди. Возникнув, пожар часто в короткие сроки распространяется на многие десятки километров. Борьба с ним очень тяжело. Космические методы слежения за пожарами помогают выявлять их на начальной стадии и тем самым способствуют оперативности принятия мер, позволяют осуществлять мониторинг и оценивать последствия. Важно обнаруживать пожары на начальной стадии их развития с целью минимизации материальных затрат на ликвидацию очага возгорания [1–4].

Использование космических данных для мониторинга пожарной обстановки по сравнению с аэрофотосъемкой позволяет быстро и экономически более эффективно получать объективную и независимую информацию для оперативного принятия решений по борьбе со стихией. Однако аэрофотосъемка также используется для обнаружения небольших пожаров.

Присутствие очага горения определяется по наличию в видимой части спектра основного дешифровочного признака лесных пожаров – дымового шлейфа – на исходном космическом снимке. По форме на снимке очаг напоминает конус светло-серого цвета. При распознавании дымовых шлейфов пожаров следует помнить, что они могут по своей структуре и яркости напоминать перистую и слоистую облачность. Поэтому те снимки, где в видимой части спектра предварительно обнаружен пожар, просматриваются в инфракрасной зоне. На тепловых инфракрасных снимках шлейфы дыма от лесных пожаров практически не видны, а очаги хорошо локализируются как выделяющиеся по яркости горячие объекты. Сочетание съёмки в видимом и тепловом инфракрасном диапазонах увеличивает возможности мониторинга лесных пожаров.

Данные мониторинга пожаров являются контекстной оценкой, на степень точности которой влияют многие факторы. В первую очередь, сплошная облачность. Очаги (термоточки, тепловые аномалии) с высокой степенью вероятности детектируются при безоблачной или малооблачной погоде (см. потеря видимости пожаров со спутника из-за облачности). Кроме того, на вероятность обнаружения пожара влияют такие факторы, как разностная температура и размер пожара (см. минимальный размер детектируемых пожаров) [3].

На практике существует ряд подходов к детектированию пожаров. Так, в статье [5] детектирование осуществляется на основе сегментирования движущихся объектов, что позволило достичь точности более 96 %. Однако данный подход требует последовательности снимков одного участка поверхности Земли.

Авторы [6] предлагают алгоритм обработки снимков геостационарного спутника Himawari-8, позволяющий снизить количество ложных сигналов, повышая точность детектирования до 93 %.

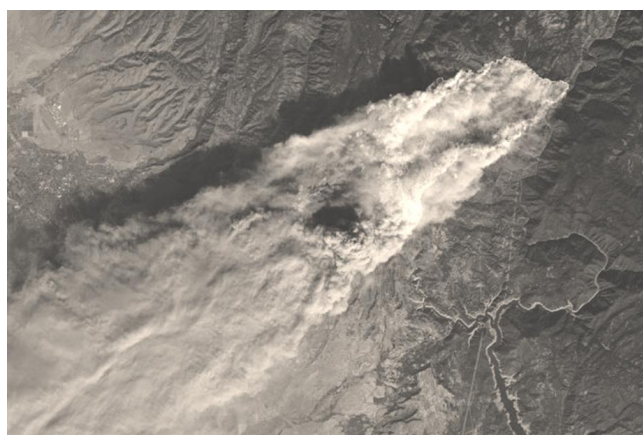
Используются для данной задачи и нейронные сети. Так, авторы [7] представили классификатор на основе сверточных нейронных сетей, позволивший достигнуть точности 90 % на валидационной выборке. Классификатор на основе глубокой сверточной сети с точностью до 98 % на данных аэрофотосъемки описан и в [8].

**Постановка задачи.** Задачей исследования является разработка алгоритма преобразования  $A: I_{orig} \rightarrow I_{result}$ , позволяющего получить из исходного изображения поверхности Земли  $I_{orig}$  такое

изображение  $I_{result}$ , каждый пиксель  $I_{result}(x, y)$  которого получен из соответствующего пикселя изображения  $I_{orig}(x, y)$  либо его окрестности. Каждый пиксель изображения  $I_{orig}$  соответствует точке в пространстве RGB. Каждый пиксель  $I_{result}(x, y)$  соответствует одному из классов («нормальная поверхность» и «пожар»).

Материалами для исследований явились спутниковые изображения поверхности Земли, содержащие пожары.

Основным признаком является наличие дыма и, в некоторых случаях, открытого пламени, который может быть замечен при спутниковой съемке.



**Рисунок 1** – Примеры исходного снимка, содержащего пожар

**Подготовка данных для обучения и валидации.** Обучающая выборка была получена путем «нарезки» имеющихся снимков с размеченными участками. При этом нарезались участки размером 384×384 пикселей с перекрытием, отражением по вертикали и горизонтали, а также с добавлением поворотов на углы, кратные 90°. Маска классов представляет собой полутоновое изображение, соответствующее по размеру с фотоснимком. Изображение маски содержит количество уровней яркости, равное количеству классов на изображении. При этом классам соответствуют следующие значения яркости: 0 – «нормальная поверхность», 1 – «пожар».

**Сегментатор на базе SegNet.** Сегментатор на базе SegNet  $A_s$  представляет собой СНС [9, 10], осуществляющую сегментацию на три сегмента: «нормальная поверхность» и «пожар».

Эмпирически выбраны следующие параметры СНС:

- Размер входного слоя 384×384×3 (цветное изображение).
- Свечточный слой Conv2D\_1.1: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 32$ , функция активации – ReLU.
- Свечточный слой Conv2D\_1.2: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 32$ , функция активации – ReLU.
- Слой понижения размерности в 2 раза MaxPooling2D\_1.
- Свечточный слой Conv2D\_2.1: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 64$ , функция активации – ReLU.
- Свечточный слой Conv2D\_2.2: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 64$ , функция активации – ReLU.

**Ганченко Валентин Вячеславович**, к. т. н., научный сотрудник лаборатории идентификации систем Объединенного института проблем информатики Национальной академии наук Беларуси; e-mail: ganchenko@isi.bas-net.by.

**Дудкин Александр Арсентьевич**, д. т. н., профессор, зав. лабораторией идентификации систем Объединенного института проблем информатики Национальной академии наук Беларуси.

Беларусь, ОИГИ НАН Беларуси, 220012, г. Минск, ул. Сурганова, 6.

ство фильтров  $F_c = 64$ , функция активации – ReLU.

- Слой понижения размерности в 2 раза MaxPooling2D\_2.
- Свечточный слой Conv2D\_3.1: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 128$ , функция активации – ReLU.
- Свечточный слой Conv2D\_3.2: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 128$ , функция активации – ReLU.
- Слой понижения размерности в 2 раза MaxPooling2D\_3.
- Слой повышения размерности в 2 раза UpSampling2D\_1: интерполяция – билинейная.
- Свечточный слой Conv2D\_4.1: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 256$ , функция активации – ReLU.
- Свечточный слой Conv2D\_4.2: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 256$ , функция активации – ReLU.
- Слой повышения размерности в 2 раза UpSampling2D\_2: интерполяция – билинейная.
- Свечточный слой Conv2D\_5.1: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 128$ , функция активации – ReLU.
- Свечточный слой Conv2D\_5.2: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 128$ , функция активации – ReLU.
- Слой повышения размерности в 2 раза UpSampling2D\_3: интерполяция – билинейная.
- Свечточный слой Conv2D\_6.1: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 64$ , функция активации – ReLU.
- Выходной свечточный слой Conv2D\_6.2: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 2$ , функция активации – сигмоидная, размер выходного слоя  $384 \times 384 \times 2$ .

Функция потерь (loss) – softmax cross entropy [11].

Обучение:

- Размер обучающей выборки: 4000 изображений
- Размер валидационной выборки: 800 изображений.
- Результирующая точность на валидационной выборке: 91,99 %.

**Сегментатор на базе U-Net.** Сегментатор на базе U-Net  $A_u$  представляет собой СНС, осуществляющую сегментацию на три сегмента: «нормальная поверхность» и «пожар». Данная архитектура отличается от SegNet наличием дополнительных связей между слоями свертки, что технически выражается добавлением слоев конкатенации.

Эмпирически выбраны следующие параметры СНС:

- Размер входного слоя  $384 \times 384 \times 3$  (цветное изображение).
- Свечточный слой Conv2D\_1.1: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 32$ , функция активации – ReLU.
- Свечточный слой Conv2D\_1.2: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 32$ , функция активации – ReLU.
- Слой понижения размерности в 2 раза MaxPooling2D\_1.
- Свечточный слой Conv2D\_2.1: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 64$ , функция активации – ReLU.
- Свечточный слой Conv2D\_2.2: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 64$ , функция активации – ReLU.
- Слой понижения размерности в 2 раза MaxPooling2D\_2.
- Свечточный слой Conv2D\_3.1: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 128$ , функция активации – ReLU.
- Свечточный слой Conv2D\_3.2: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 128$ , функция активации – ReLU.
- Слой понижения размерности в 2 раза MaxPooling2D\_3.
- Слой повышения размерности в 2 раза UpSampling2D\_1: интерполяция – билинейная.
- Слой конкатенации слоев UpSampling2D\_1 и Conv2D\_3.2.
- Свечточный слой Conv2D\_4.1: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 256$ , функция активации – ReLU.
- Свечточный слой Conv2D\_4.2: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 256$ , функция активации – ReLU.
- Слой повышения размерности в 2 раза UpSampling2D\_2: интерполяция – билинейная.

- Слой конкатенации слоев UpSampling2D\_2 и Conv2D\_2.2.
  - Свечточный слой Conv2D\_5.1: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 128$ , функция активации – ReLU.
  - Свечточный слой Conv2D\_5.2: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 128$ , функция активации – ReLU.
  - Слой повышения размерности в 2 раза UpSampling2D\_3: интерполяция – билинейная.
  - Слой конкатенации слоев UpSampling2D\_3 и Conv2D\_1.2.
  - Свечточный слой Conv2D\_6.1: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 64$ , функция активации – ReLU.
  - Выходной свечточный слой Conv2D\_6.2: размер фильтра  $F_s = 3$ , количество фильтров  $F_c = 2$ , функция активации – сигмоидная, размер выходного слоя  $384 \times 384 \times 2$ .
- Функция потерь (loss) – softmax cross entropy.

Обучение:

- Размер обучающей выборки: 4000 изображений
- Размер валидационной выборки: 800 изображений.
- Результирующая точность на валидационной выборке: 93,20 %.

Структура выходных данных. В качестве выхода реализованных СНС выступает матрица размером  $384 \times 384 \times 2$ , где размеры « $384 \times 384$ » соответствуют размеру входного изображения, а «2» – количеству искомым классов: «нормальная поверхность» и «пожар». Таким образом, на выходе получается три матрицы значения, в которых соответствуют вероятности принадлежности пикселя исходного изображения к тому или иному классу. Выполнив нормализацию значений для каждого пикселя получаем нечеткую величину, характеризующую принадлежность пикселя к искомым классам.

**Алгоритм распознавания.** В целом алгоритм распознавания (преобразование  $A: I_{orig} \rightarrow I_{result}$ ) на базе того или иного реализованного сегментатора можно представить в следующем виде:

1. Загрузить исходное цветное изображение  $I_{orig}$  размером  $x \times N$ .

2. Исходное изображение  $I_{orig}$  разделить на области размером  $384 \times 384$  пикселей. Для каждой области:

2.1. Копируем выбранную область  $i(I_{orig})$  размером  $384 \times 384$  пикселей в виде цветного изображения.

2.2. Подаем полученное изображение  $i(I_{orig})$  на сегментатор  $A_s$  или  $A_u$ .

2.3. В случае пересечения областей соответствующие степени принадлежности пикселя  $(x, y)$  усредняются.

2.4. Получаем индекс класса для каждого пикселя изображения  $i(I_{orig})(x, y): \in [0, 255], y \in [0, 255],$

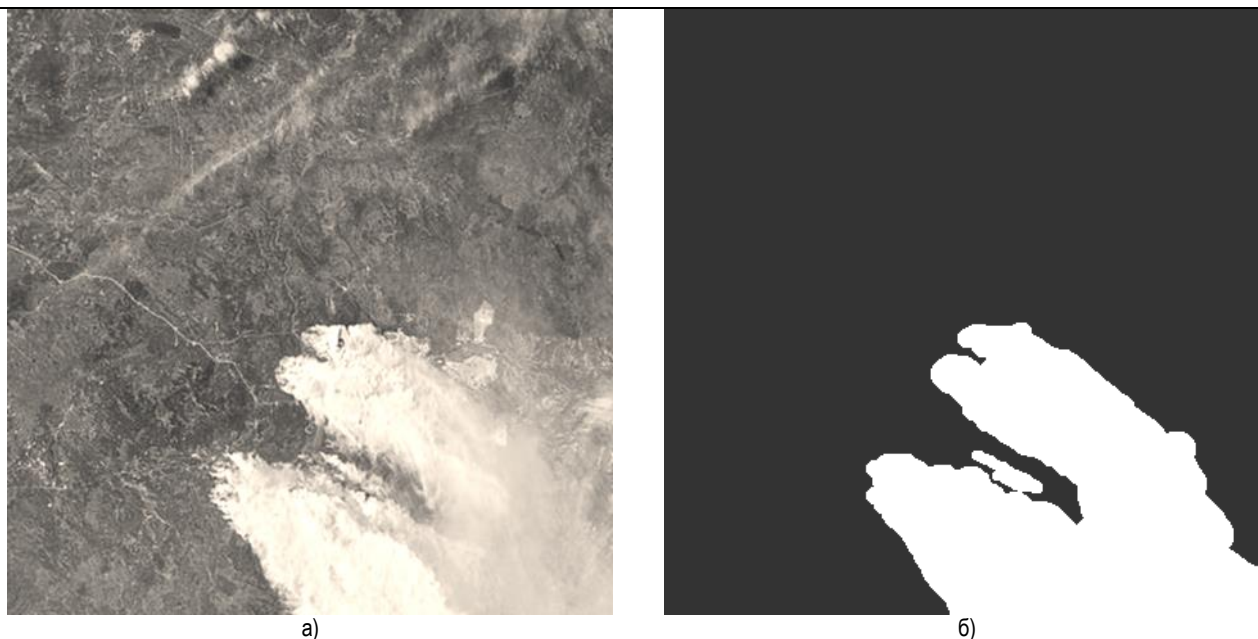
$$index = \text{argmax}([A(x, y)]),$$

где  $Segm_A(x, y)$  – вектор из трех значений соответствующих степени принадлежности к искомым классам в точке  $(x, y)$  исходного изображения  $i(I_{orig})$  полученного с помощью выбранного сегментатора ( $A_s$  или  $A_u$ ).

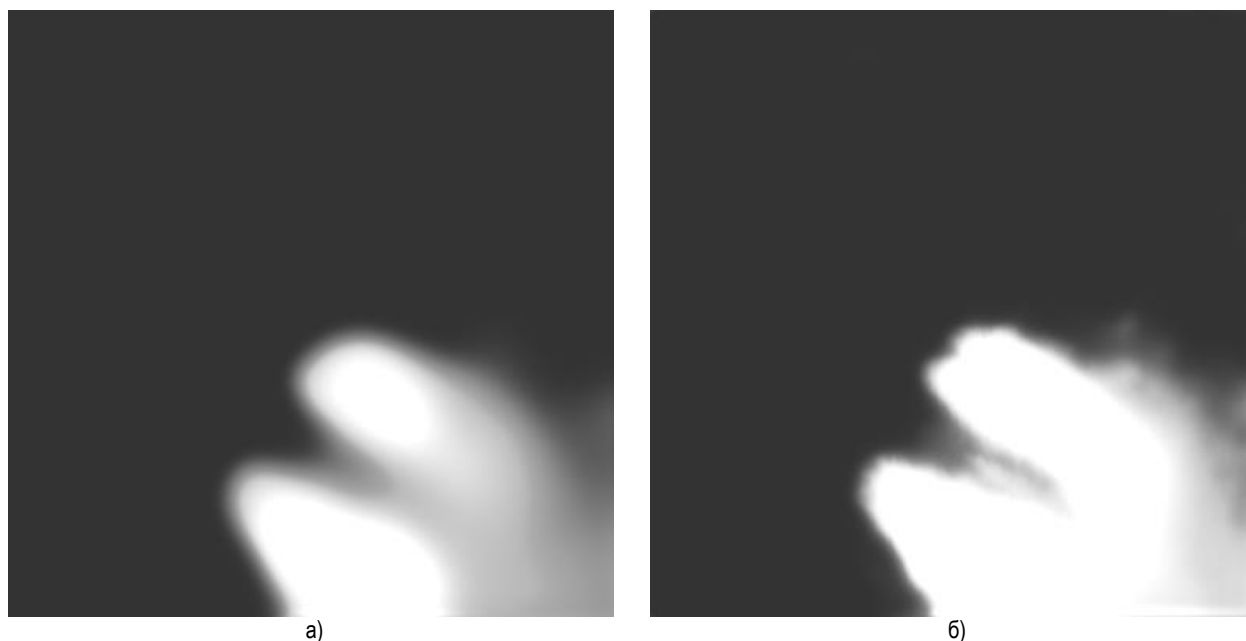
2.5. Присваиваем пикселю выходного изображения области  $I_{result}(O_i)$  значение, соответствующее псевдоцвету индекса класса: черный – нормальная поверхность, белый – пожар.

3. Собираем и сохраняем полученное изображение  $I_{result}$ .

**Тестирование.** Тестирование сегментаторов осуществлялось на валидационной выборке. При этом оценивалась точность как для каждого класса в отдельности, так и для всех классов в целом. Полученные результаты тестирования приведены в таблице 1.



а) б)  
Рисунок 2 – Пример участка исходного снимка (а) и размеченный класс (б)



а) б)  
Рисунок 3 – Полученные в результате с помощью SegNet (а) и U-Net (б) степени принадлежности точек участка сегментированного изображения классам

Таблица 1 – Результаты тестирования сегментаторов

Классы	Точность, %	
	SegNet	U-Net
Нормальная поверхность	97,90	93,72
Пожар	65,74	90,89
<b>Общая точность</b>	<b>91,99</b>	<b>93,20</b>

Из-за несбалансированности классов в исходных данных требуется дополнительная оценка. Данные, полученные в результате, сведены в матрицы неточностей (confusion matrix), приведенные в таблице 2. Значения в матрицах приведены в относительных величинах – количество пикселей, принадлежащих классу, к общему количеству пикселей всех классов в выборке.

Таблица 2 – Матрицы неточностей сегментаторов

Предсказанные классы	Реальные классы			
	SegNet		U-Net	
	Норма	Пожар	Норма	Пожар
Норма	79,93	1,79	76,52	1,67
Пожар	6,29	12,07	5,12	16,69

Для оценки качества работы сегментаторов были вычислены соответствующие значения точности (precision), полноты (recall) и  $F_1$ -меры [12] (см. табл. 3).

Таблица 3 – Значения точности (precision), полноты (recall) и  $F_1$ -меры

Классы	SegNet			U-Net		
	Precision	Recall	$F_1$	Precision	Recall	$F_1$
Норма	0.93	0.98	0.95	0.98	0.94	0.96
Пожар	0.88	0.66	0.75	0.77	0.91	0.83

Наибольшее количество ошибок возникало на участках, соответствующих границе задымления, а также в случае значительного рассеивания дыма.

На рисунке 2 приведен пример участка исходного изображения и соответствующей разметки классов.

На рисунке 3 изображены степени принадлежности точек участка сегментированного изображения классам. Для сравнения приведены размеченные экспертом классы.

**Заключение.** Для обработки спутниковых снимков предложены и реализованы с использованием библиотеки Keras (в качестве бэкенда использовалась библиотека Tensorflow) семантические сегментаторы. Предложенные сегментаторы построены по архитектурам SegNet и U-Net и обучены на двух классах: «нормальная поверхность» и «пожар». При использовании предложенных сегментаторов удалось достичь точности 92–93 %. При этом наибольшее количество ошибок возникает при рассеивании задымления.

Дальнейшие исследования предполагают использования transfer learning на базе обученных сетей архитектур VGG и ResNet для снижения количества ошибок в проблемных областях.

*Работа выполнена при поддержке БРФФИ (проект №Ф18В-005) и ГКНТ Республики Беларусь (проект №Ф18ПШГ-008П).*

#### СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Использование космических снимков в тепловом инфракрасном диапазоне для географических исследований [Электронный документ]. – Режим доступа : [http://www.geogr.msu.ru/cafedra/karta/materials/heat\\_img/files/2/pozhary.htm](http://www.geogr.msu.ru/cafedra/karta/materials/heat_img/files/2/pozhary.htm). – Дата доступа : 12.01.2020.
2. Карпов, А. А. Технологии определения природных пожаров с использованием данных спутниковой съемки / А. А. Карпов, Р. А. Алешко, К. В. Шошина // Молодой ученый. – 2015. – № 13(1) – С. 17–19.

3. СКАНЭКС – мониторинг природных пожаров, карта пожаров [Электронный ресурс]. – Режим доступа : <https://fires.ru/help.html>. – Дата доступа : 12.01.2020.
4. Early Forest Fire Detection and Verification using Optical Smoke, Gas and Microwave Sensors / Wolfgang Krüll, Robert Tobera, Ingolf Willms, Helmut Essen, Noravon Wahl // Procedia Engineering. – 2012. – Vol. 45. – P. 584–594.
5. Mahmoud, Mubarak A. I. Forest Fire Detection Using a Rule-Based Image Processing Algorithm and Temporal Variation / Mubarak A. I. Mahmoud, Honge Ren // Mathematical Problems in Engineering. – 2018. – Vol. 2018. – 8 p.
6. Detection and Monitoring of Forest Fires Using Himawari-8 Geostationary Satellite Data in South Korea / Eunna Jang, Yoojin Kang, Jung-ho Im, Dong-Won Lee, Jongmin Yoon, Sang-Kyun Kim // Remote Sens. – 2019. – Vol. 11(3). – 25 p.
7. Zhang, Q. Deep Convolutional Neural Networks for Forest Fire Detection / Qingjie Zhang, Jiaolong Xu, Liang Xu, Haifeng Guo // Proc. of International Forum on Management, Education and Information Technology Application. – 2016.
8. Zhao, Y. Saliency Detection and Deep Learning-Based Wildfire Identification in UAV Imagery / Yi Zhao, Jiale Ma, Xiaohui Li, Jie Zhang // Sensors (Basel). – 2018. – Vol. 18(3). – 19 p.
9. Goodfellow, I. Deep learning / I. Goodfellow, Y. Bengio, A. Courville // The MIT Press. – 2016. – 800 p.
10. Николенко, С. Глубокое обучение / С. Николенко, А. Кадурич, Е. Архангельская – СПб. : Питер, 2018. – 480 с.
11. Tensorflow API documentation [Электронный ресурс]. – Режим доступа : [https://www.tensorflow.org/api\\_docs/python/tf/nn/softmax\\_cross\\_entropy\\_with\\_logits\\_v2](https://www.tensorflow.org/api_docs/python/tf/nn/softmax_cross_entropy_with_logits_v2). – Дата доступа : 28.09.2018.
12. Sokolova, M. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation / Marina Sokolova, Nathalie Japkowicz, Stan Szpakowicz // Advances in Artificial Intelligence, 19th Australian Joint Conference on Artificial Intelligence. – 2006. – Hobart, Australia, December 4-8, 2006. – P. 1015–1021.

26.02.2020

#### GANCHENKO V. V., DOUDKIN A. A. Semantic segmentation in tasks of fire detection on data of remote sensing of the earth

In this paper, we consider problem of detection of fires according to data of remote sensing of the Earth's surface of various spatial resolutions. As a basis for detection, semantic segmentation is used. This segmentation approach is based on convolutional neural networks. The results of two neural network architectures based on SegNet and U-Net are compared.

004.89

• • • •

**Введение.** В настоящее время нейронные сети широко применяются для решения различного рода прикладных задач. При этом наиболее сложным этапом нейросетевой обработки данных является процесс обучения сети [1, 2]. Несмотря на достигнутые в данном направлении результаты, проблема обучения по-прежнему является актуальной [3].

Существуют два основных подхода к обучению нейронных сетей. Один базируется на методологии градиентного спуска, а другой – на случайном поиске. Наиболее популярными являются градиентные методы, которые на практике обучают нейронные сети за приемлемое время [4]. Представители другого подхода – методы отжига – обеспечивают хорошее качество, но работают существенно медленнее [5].

В работе изучаются возможности метода отжига для обучения глубокой доверительной сети на примере решения задачи сжатия цветных изображений.

**1. Анализ проблемы.** Глубокие доверительные сети использу-

ются для решения целого ряда прикладных задач, таких как медицинская диагностика, распознавание образов, обработка изображений, выделение семантически значимых признаков и т. д. [6].

Опишем архитектуру глубокой доверительной сети. Для этого вначале надо описать архитектуру ограниченной машины Больцмана, так как любая глубокая доверительная сеть всегда содержит слои данного типа машины.

Известно, что в основе ограниченной машины Больцмана лежит стохастический нейрон.

Формально ее можно представить полносвязным двудольным графом  $G=(X, U)$ ,

$$\begin{cases} X = X_1 \cup X_2, X_1 \cap X_2 = \emptyset \\ U = \{u = (x_1, x_2) \mid \forall x_1 \in X_1, \forall x_2 \in X_2\}, \end{cases} \quad (1)$$

где  $X$  – множество вершин – стохастических нейронов,  $U$  – множество

*Краснопрошин Виктор Владимирович, д. т. н., профессор, зав. кафедрой информационных систем управления Белорусского государственного университета.*

*Мацкевич Вадим Владимирович, магистрант кафедры информационных систем управления Белорусского государственного университета.*

*Беларусь, БГУ, 220050, г. Минск, пр. Независимости, 4.*