Распознавание текста на зашумленных изображениях

А.А. Рыбаков[1], С.А. Фрейлехман[2]

[1]МСЦ РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия, rybakov@jscc.ru;

[2] МСЦ РАН – филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия, freysa@jscc.ru, +7 926 125-66-20

Аннотация. Автоматическая сегментация изображений является востребованной задачей. В современном мире эта задача решается с использованием обученных искусственных нейронных сетей. Для обучения нейросети необходимо подготовить обучающий набор данных. Подобный набор данных должен состоять из изображений для обучения и «правильных ответов» к ним. Формирование набора данных является трудоемкой и дорогостоящей задачей. Если учитывать необходимое количество этих пар (сотни или тысячи штук), то задача может стать трудно выполнимой за разумное время. Для решения этой задачи было предложено разработать генератор обучающей выборки.

Ключевые слова: генератор текста, искусственная нейронная сеть, машинный текст, машинное обучение

1. Введение

Искусственные нейронные сети (ИНС) структурируют и анализируют большие объемы данных. Все чаще применяется машинное обучение для предсказания событий и обнаружения неочевидных закономерностей. Все это помогает создавать и улучшать продукты в науке, промышленности и бизнесе [1], [2].

Одной из популярных задач в области классификации объектов с использованием ИНС является задача распознавания текста.

Сложность обучения ИНС для решения этой задачи, как и во многих других задачах, сводится к подготовке качественного набора обучающих данных. Данное суждение основано на том, что насколько бы ни была сложной или наоборот, простой архитектура модели ИНС, она не сможет решить поставленную перед ней задачу, если она не обладает качественной тренировочной выборкой, на которой она сможет учиться [3].

В данной статье приводится анализ классических моделей ИНС для решения задачи распознавания текста с использованием генератора обучающей выборки.

2. Постановка задачи исследования

Автоматическая сегментация изображений (*image segmentation*) – востребованное направление использования нейронных сетей. Например, чтобы вручную разметить спутниковые снимки, обрисовав контуры домов, дорог, парковых зон или выделить текст на фоне некоего изображения, нужно потратить колоссальное количество времени.

Подобные задачи ныне решаются с использованием ИНС. Для этого ИНС на вход подается обучающая выборка, состоящая из исходного исследуемого изображения и сегментированного варианта исходного изображения, преобразованного в маску. Результатом обучения станет модель ИНС с подобранными весами, способная решать поставленную перед ней задачу с определенной долей погрешности, от которой собственно и зависит качество самой модели [4].

В данной статье исследуется задача бинарной сегментации изображения с черным машинным текстом на случайном фоне. Размер текста – крупный, для того, чтобы при сжатии изображения до входного тензора ИНС сохранились основные признаки символов текста. Результатом работы модели ИНС ожидается классификация пикселей исходного изображения на два класса - текст и фон (не текст).

Метод исследования – попиксельная сегментация. В данном методе каждому пикселю исходного изображения модель ИНС будет присваивать один из двух классов. Реализация данного метода заключается в формировании обученной моделью маски изображения со значениями пикселей 1 и 0, где: 1 – текст, 0 – фон (не текст).

Цель исследования: анализ методов получения эффективного набора данных для обучения модели ИНС для задачи бинарной сегментации изображения с текстом.

Задачи исследования:

- получить некий первичный набор данных для исследований с использованием генератора данных;
- оценить характерные свойства полученной

выборки;

- задать параметры обучения модели;
- экспериментальным путем отобрать подходящую модель ИНС для решения данной задачи;
- применительно к выбранной модели провести анализ влияния параметров обучающей выборки на качество предсказаний модели.

2.1. Основные параметры обучающего набора данных

В рамках данного исследования было принято решение исследовать изображения, содержащие *машинный текст на русском языке* (кириллица). Также модель необходимо научить распознавать символы и знаки, часто встречающиеся в русских текстах, для полноты передачи исходного текста после обработки изображения моделью ИНС.

Полностью русскоязычный алфавит с дополнительными символами для генератора обучающих данных представлен ниже:

```
alphabet_RU = 'абвгдеёжзийклмнопрстуфхцчшщъыьэюяАБВГДЕЁЖЗИКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ'
symbol_RU = '!@#$%*+/=()[]{}~"″№<>,.;:?'
```

Для генерируемой обучающей выборки было принято решение в качестве *цвета текста* использовать случайные *оттенки черного*.

Для выборки *фон* генерируется из *случайных фигур случайного цвета*.

В рамках данной задачи подразумевается, что изображение, содержащее текст, не обязательно должно иметь идеально горизонтальное расположение строк. Потому введена возможность *случайного поворота изображения с текстом* в пределах заданных углов (± 5º).

В данной задаче хоть и исследуется только *машинный текст*, но и он бывает представлен в виде *различных стилей и размеров шрифтов*. Для определения этого параметра исходных обучающих данных для модели ИНС, были отобраны наиболее часто встречающиеся шрифты [5], [6].

Ниже представлен список шрифтов, которые применялись для генерации текста на обучающей выборке:

```
often_fonts = ['arial.ttf',
'ARIALN.TTF', 'BOOKOS.TTF',
'cour.ttf', 'GOTHIC.TTF',
'georgia.ttf', 'lucon.ttf',
'micross.ttf', 'verdana.ttf',
'ariblk.ttf', 'BKANT.TTF',
'comic.ttf', 'GARA.TTF', 'impact.ttf',
'l_10646.ttf', 'times.ttf',
'lucon.ttf']
```

Пример сгенерированных дынных, с учетом выше предложенных переменных, представлен на рисунке 1.

Изображение Маска текста

Рис. 1. Пример работы генератора

2.2. Свойства генератора обучающей выборки

Чтобы оценить качество обучающей выборки, состоящей из набора параметров, определенных случайным образом, введем понятие совокупности общих свойств, которые будут оказывать прямое влияние на формирование весов модели – формирование признаков, по которым модель будет распознавать текст.

Представленные выше условия формирования генератора и сам генератор, формируют набор данных со следующими свойствами:

- случайный фон из всей палитры цветов (имеет однозначные границы перехода цветов – нет плавного перехода цветов);
- цвет текста – оттенки черного в диапазоне от 0 до 30 по каждому каналу (формат цвета RGB);
- текст разного размера;
- текст имеет разные шрифты;
- тест имеет случайный угол наклона;
- из-за того, что картинки генерируются разных размеров, а на вход в модель подаются меньшего размера, то появляются шумы, которые также становятся качествами – свойствами данной выборки.

Отсюда можно предположить какую совокупность признаков сможет сгенерировать модель для распознания текста на изображении:

- геометрические очертания букв и символов теста, отличные от фона;
- резкий перепад градиент цвета пикселей на границе текста и фона;
- цвет фона и текста - различаются.

3. Параметры модели сегментации изображения

3.1. Гиперпараметры для обучения модели ИНС

Перед обучением ИНС необходимо настроить гиперпараметры – параметры, которые зада-

ются перед обучением и не являются результатом вычислений ИНС. В данной статье к таким параметрам отнесены: размер сгенерированного датафрейма, его тренировочная и обучающая части, размер пакета данных, подаваемого за раз на обучение, количество шагов в одной эпохе, количество эпох обучения, количество шагов валидации и значение шагов через которое необходимо проводить валидацию результатов обучения. Значения вышеописанных параметров:

```
dataframe = 1000
train_dataframe = 700
val_dataframe = 300
batch_size = 20
steps_per_epoch = 100
epochs = 25
validation_steps = 50
validation_freq = 1
```

Прочие не указанные гиперпараметры, например, такие как скорость обучения и параметры ее изменения во время обучения, взяты по умолчанию из библиотеки Keras [7].

3.2. Размерность входного и выходного тензоров модели

Тензор входного слоя для всех моделей – (batch_size, 256, 256, 3) или (*None, 256, 256, 3*) на момент компиляции модели;

Размер тензора выходного слоя моделей при решении задачи сегментации изображения (классификации пикселя) – (*None, 256, 256, 1*) на момент компиляции модели.

3.3. Настройки модели

Функция активации нейронов – *ReLu*. ReLu (рис. 2) менее требовательно к вычислительным ресурсам, чем гиперболический тангенс или сигмоида, так как производит более простые математические операции.

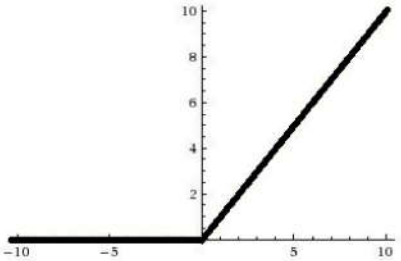


Рис. 2. Функция активации ReLu

Данная функция используется для активации скрытых слоев в каждой исследуемой модели.

В качестве выходной функции активации используется *сигмоида* (рис. 3) – логистическая функция активации, результатом работы которой является вероятность принадлежности к анализируемому классу. Ее диапазон значений от 0 до 1.

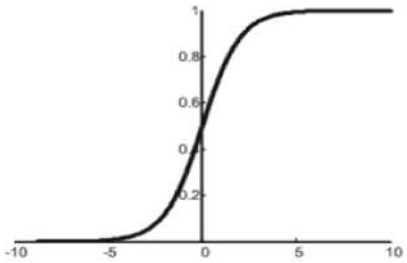


Рис. 3. Функция активации выходного слоя

Данная функция используется для активации выходных нейронов в каждой исследуемой модели.

Функция оптимизации модели для достижения глобальных минимумов во время обучения модели – *Adam*, так как он выдаёт наилучшие результаты при минимальном подборе параметров [7]:

```
Adam(lr=0.0001, beta_1=0.9,
     beta_2=0.999, epsilon=1e-08,
     decay=0.0)
```

Обучающая функция потерь (*loss*) – двоичная кросс-энтропия (binary_crossentropy), как наиболее подходящая функция потерь при тренировке двоичного классификатора [8].

3.4. Исследуемые модели ИНС

Сверточные нейронные сети (convolutional neural networks, CNN) и *глубинные сверточные нейронные сети* (deep convolutional neural networks, DCNN) сильно отличаются от других видов сетей. Обычно они используются для обработки изображений, реже для аудио. Типичным способом применения CNN является классификация изображений. Для формирования данной архитектуры за основу использовали некоторые особенности зрительной коры, в которой были открыты так называемые простые клетки, реагирующие на прямые линии под разными углами, и сложные клетки, реакция которых связана с активацией определенного набора простых клеток. Таким образом, идея сверточных нейронных сетей заключается в чередовании сверточных слоев (англ. convolution layers) и субдискретизирующих слоев (англ. subsampling layers или англ. pooling layers, слоёв подвыборки). Структура сети – однонаправленная (без обратных связей), принципиально многослойная [9].

SegNet – это архитектура глубокого кодировщика-декодера для мультиклассовой пиксельной сегментации, также эффективная для понимания сцен внутри помещений, исследованная и разработанная членами группы компьютерного зрения и робототехники Кембриджского университета, Великобритания.

Архитектура состоит из последовательности уровней нелинейной обработки (кодировщиков) и соответствующего набора декодеров, за которыми следует пиксельный классификатор. Обычно каждый кодер состоит из одного или нескольких сверточных слоев с пакетной нормализацией и нелинейностью ReLU, за которыми следуют неперекрывающиеся maxpooling и субдискретизация. Разреженное кодирование из-за процесса объединения повышается в декодере с использованием индексов maxpooling в последовательности кодирования (рис. 4). Одним из ключевых компонентов SegNet является использование индексов максимального объединения в декодерах для выполнения повышающей дискретизации карт функций с низким разрешением.

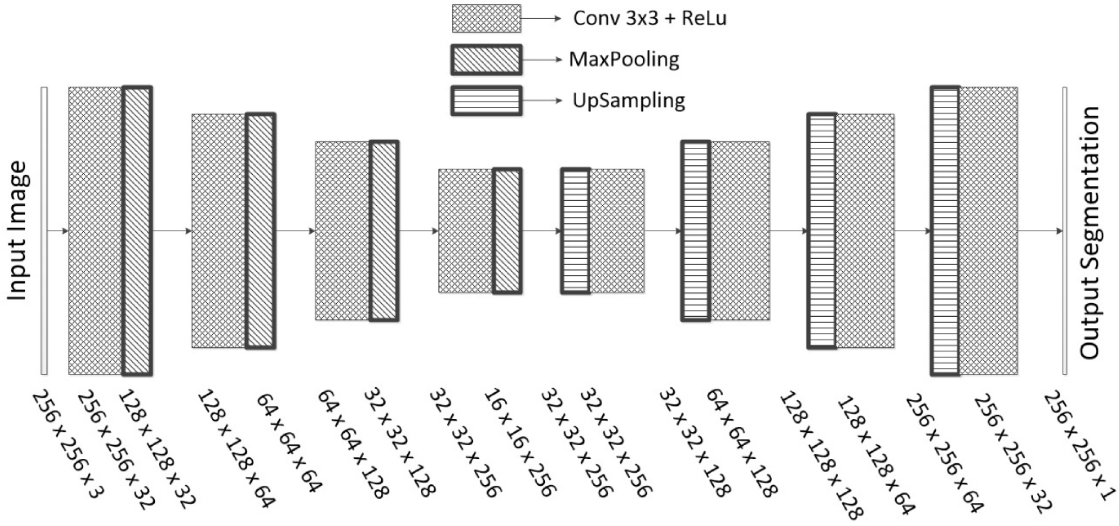


Рис. 4. Архитектура модели SegNet

Это имеет важные преимущества, заключающиеся в сохранении высокочастотных деталей в сегментированных изображениях, а также в сокращении общего количества обучаемых параметров в декодерах [10].

U-Net считается одной из стандартных архитектур CNN для задач сегментации изображений, когда нужно не только определить класс изображения целиком, но и сегментировать его области по классу, т. е. создать маску, которая будет разделять изображение на несколько классов.

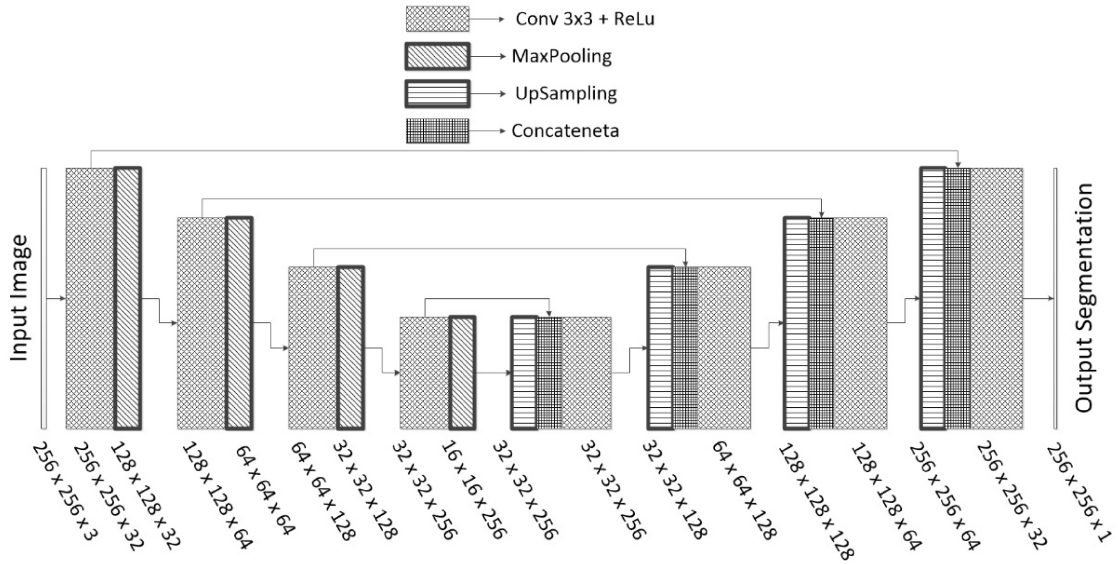


Рис. 5. Архитектура модели U-Net

Архитектура состоит из стягивающего пути

для захвата контекста и симметричного расширяющегося пути, который позволяет осуществить точную локализацию.

Для U-Net характерно:

- достижение высоких результатов в различных реальных задачах, особенно для биомедицинских приложений;
- использование небольшого количества данных для достижения хороших результатов.

Архитектура ИНС (рис. 5) состоит из сужающегося пути (слева) и расширяющегося пути (справа). Сужающийся путь – типичная архитектуре сверточной нейронной сети. Он состоит из повторного применения двух сверток 3×3, за которыми следуют ReLU и операция максимального объединения (2×2 степени 2) для понижения разрешения.

На каждом этапе понижающей дискретизации каналы свойств удваиваются. Каждый шаг в расширяющемся пути состоит из операции повышающей дискретизации карты свойств, за которой следуют:

- свертка 2×2, которая уменьшает количество каналов свойств;
- объединение с соответствующим образом обрезанной картой свойств из стягивающегося пути;
- две 3×3 свертки, за которыми следует ReLU.

Обрезка необходима из-за потери граничных пикселей при каждой свертке.

На последнем слое используется свертка 1×1 для сопоставления каждого 64-компонентного вектора свойств с желаемым количеством классов. Всего сеть содержит 23 сверточных слоя [11].

4. Результат обучения ИНС

Модели SegNet и U-Net были обучены и протестированы на наборе данных, полученном с использование генератора. Все модели тренировались на одной и той же сгенерированной выборке данных.

Исходные гиперпараметры ИНС перед обучением – идентичные.

Обучение моделей производилось в облачной среде для разработки и обучения ИНС – Google Colab [12]. Обучение производилось с использованием графической карты NVIDIA Tesla P100.

Результаты обучения ИНС представлены на рисунках 6-7.

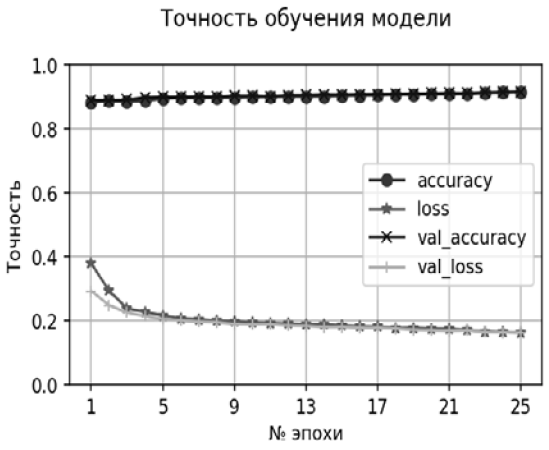


Рис. 6. Точность обучения модели SegNet

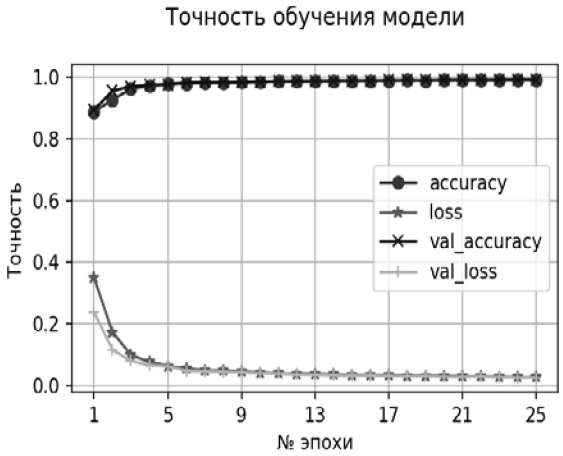


Рис. 7. Точность обучения модели U-Net

На рис. 6 и рис. 7 представлены следующие величины:

loss – ошибка обучения;
accuracy – точность обучения;
val_loss – ошибка валидации;
val_accuracy – точность валидации.

Сравнительный анализ результатов исследований приведен в таблицах 1-2.

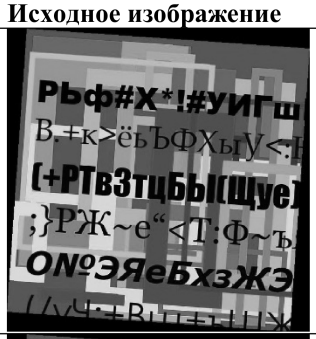
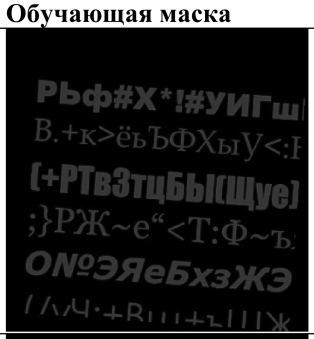
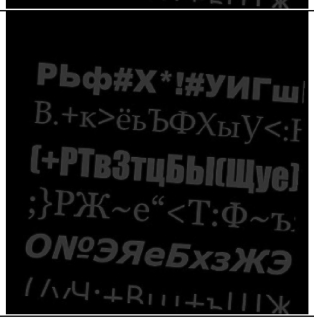
Таблица 1. Время обучения моделей ИНС

ИНС	**Время**
SegNet	1 ч. 58 мин. 46 сек.
U-Net	2 ч. 12 мин. 38 сек.

Таблица 2. Результаты обучения моделей ИНС

ИНС	**loss**	**acc.**	**val_loss**	**val_acc.**
SegNet	0,1627	0,9158	0,1628	0,9170
U-Net	0,0255	0,9899	0,02468	0,990

Таблица 3. Предсказания моделей ИНС

ИНС	Исходное изображение	Обучающая маска	Предсказание модели
SegNet			
U-Net			

5. Заключение

В таблице сравнительного анализа ИНС однозначно видно, что в ограниченных условиях обучения, архитектура модели ИНС U-Net продемонстрировала свои качества быстрого и высокоточного обучения, что было подтверждено в ходе данного эксперимента.

Созданный в рамках эксперимента генератор обучающей выборки продемонстрировал умеренные качества генерации обучающей выборки для задачи обучения ИНС сегментации изображения с черным машинным текстом на случайном фоне.

Следующими этапами разработки генератора обучающей выборки будут:

- использование произвольных цветовых схем для текста и сгенерированного фона;
- усложнение фона – добавление внутреннего генератора шумов;
- добавление возможности формирования фона с использованием естественных изображений (изображения неба, воды, ландшафтов и др.);
- добавление возможности использования различных настроек освещения изображения и размытия текста.

Также, необходимо разработать «фильтр», через который данные будут подаваться в модель ИНС. Такое решение позволит подготавливать большие изображения с малым размером шрифта текста к последующей сегментации. Это необходимо для того, чтобы не ухудшалось качество исходного изображения при его сжатии до формы входного тензора модели ИНС.

Работа выполнена в МСЦ РАН в рамках государственного задания по теме 0580-2021-0016.

Text Recognition on Noisy Images

A.A. Rybakov, S.A. Freylekhman

Abstract. Automatic image segmentation is a popular task. In the modern world, this problem is solved using trained artificial neural nets. To train a neural net, it is necessary to prepare a training data set. Such a data set should consist of images for training and "correct answers" to them. Generating a data set is a time-consuming and expensive task. If you take into account the required number of these image pairs (hundreds or thousands of them), then the task may become difficult (or not feasible). To solve this problem, it was proposed to develop a training sample generator.

Keywords: text generator, artificial neural net, machine text, machine learning

Литература

1. С.Н. Богославский. Область применения искусственных нейронных сетей и перспективы их развития. «Научный журнал КубГАУ», Т. 3 (2007), № 27, 1–11.

2. А.М. Самарин. История нейрокомпьютинга и его применение в бизнесе. «Вестник Сибирского института бизнеса и информационных технологий», Т. 6 (2013), № 2, 48-54.

3. Д.В. Васенков. Методы обучения искусственных нейронных сетей. «Компьютерные инструменты в образовании», Т. 1 (2007), № 1, 20–29.

4. С.В. Белим, С.Б. Ларионов. Алгоритм сегментации изображения с помощью искусственной нейронной сети без использования других изображений. «Радиостроение», Т. 3 (2018), № 1, 43-53.

5. Habr.com. Шрифты, общие для всех (актуальных) версий Windows, и их Mac-эквиваленты. 2009. URL: https://habr.com/ru/post/68189/ (дата обращения: 30.04.2021).

6. Redhamster. Топ-7 шрифтов, используемых профессионалами в дизайне. 2019. URL: https://redhamster.bz/top-7-shriftov-ispolzuemyh-professionalami-v-dizajne/ (дата обращения: 30.04.2021).

7. Русскоязычная документация Keras. Оптимизаторы. URL: https://ru-keras.com/optimizer/ (дата обращения: 30.04.2021).

8. Ф.М. Бетелин, А.Ф. Галимянов. Искусственные нейронные сети и их приложения. Казань, Казан. ун-та, 2018.

9. X. Zhang, J. Zhao, Y. LeCun. Character-level convolutional networks for text classification. « Proceedings of the 28th International Conference on Neural Information Processing Systems», Volume 1, 7 December, 2015, 649–657.

10. arXiv.org. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. 2017. URL: https://arxiv.org/pdf/1511.00561.pdf (дата обращения: 30.04.2021).

11. Р.А. Соловьев, Д.В. Тельпухов, А.Г. Кустов. Автоматическая сегментация спутниковых снимков на базе модифицированной свёрточной нейронной сети UNET. «Инженерный вестник дона», Т 1 (2017), № 4, 1-36.

12. Google Colaboratory. Добро пожаловать в Colaboratory!. URL: https://colab.research.google.com/notebooks/intro.ipynb (дата обращения: 30.04.2021).