УДК 004.421.2
DOI: 10.26102/2310-6018/2022.37.2.011

Синтез архитектуры нейронной сети для распознавания образов судов на базе технологии предварительного обучения

Д.И. Конарев, А.А. Гуламов✉

Юго-Западный государственный университет, Курск, Российская Федерация
profgulamov@mail.ru✉

Резюме. Актуальность статьи обусловлена инфокоммуникационным обеспечением судоходства путем мониторинга речных судов с использованием камер видеонаблюдения. Основной задачей является распознавание судов на изображениях, для чего перспективно применение нейронных сетей. Целью работы является исследование показателей эффективности распознавания судов доступными и предварительно обученными сетями после их дообучения под поставленные задачи и выбор наиболее эффективной сети. В работе рассмотрены различные предварительно обученные нейронные сети. Входными данными для сетей являются изображения судов. Обучающая выборка собрана вручную и включает в себя два независимых DataSet с изображениями речных судов и множества других объектов, за исключением судов. Сети построены и дообучены с использованием библиотек машинного обучения Keras и TensorFlow. Описано применение предварительно обученных сверточных искусственных нейронных сетей для задач распознавания образов и преимущества использования такой сети перед синтезом нейронной сети с нуля. Подробно описана архитектура эффективной предварительно обученной нейронной сети VGG16. Проведен эксперимент по дообучению доступных предварительно обученных сверточных нейронных сетей под поставленную задачу. Проведена оценка эффективности различных дообученных нейронных сетей в процентном соотношении случаев правильного распознавания образов на тестовой выборке. Выбрана наиболее эффективная нейронная сеть для задач распознавания образов судов. Максимальную точность показали сети NASNetMobile и NASNetLarge. Однако минимальный размер изображений, с которым могут работать эти сети, больше, чем для остальных доступных сетей, что увеличивает число параметров в сверточных слоях этих сетей и обусловливает существенное возрастание времени дообучения и работы, чем для остальных доступных сетей. Вместе с тем нейронная сеть VGG16 при небольшом числе параметров и малом времени на дообучение показала очень высокую эффективность, ввиду чего рекомендована к использованию для задачи распознавания образов судов.

Ключевые слова: искусственные нейронные сети, предварительно обученные сети, сверточные нейронные сети, Keras, TensorFlow, Google colaboratory, VGG16, NASNetMobile, NASNetLarge.

Для цитирования: Конарев Д.И., Гуламов А.А. Синтез архитектуры нейронной сети для распознавания образов судов на базе технологии предварительного обучения. *Моделирование, оптимизация и информационные технологии.* 2022;10(2). Доступно по: https://moitvivt.ru/ru/journal/pdf?id=1148 DOI: 10.26102/2310-6018/2022.37.2.011

Synthesis of neural network architecture for ship pattern recognition based on pre-training technology

D.I. Konarev, A.A. Gulamov✉

South-West State University, Kursk, Russian Federation
profgulamov@mail.ru✉

Abstract. The relevance of the article is due to the information and communication support of navigation by monitoring river vessels using video surveillance cameras. The main goal is to recognize ships in

images, for which the application of neural networks has potential. The aim of the paper is to study the performance indicators of vessel recognition by means of available pre-trained networks after their additional training for the assigned tasks and to select the most efficient network. The research considers various pre-trained neural networks. The input data for the networks are ship images. The training sample was collected manually and includes two independent DataSets with images of river vessels and many other objects apart from ships. The networks were built and further trained with the aid of Keras and TensorFlow machine learning libraries. The employment of pre-trained convolutional artificial neural networks for pattern recognition problems and the advantages of utilizing such networks over synthesizing a neural network from scratch are presented. The architecture of efficient pre-trained VGG16 neural network is described in detail. An experiment was conducted in additional training of available pre-trained convolutional neural networks for the assigned task. The efficiency of various pre-trained neural networks was evaluated in terms of the percentage of correct pattern recognition cases on the test set. The most efficient neural network for ship pattern recognition tasks has been selected. NASNetMobile and NASNetLarge networks have shown the maximum accuracy. However, the minimum image size that these networks can work with is larger than for other available networks and the great number of parameters in the convolutional layers of these networks causes a significant increase in retraining and operation time than for other available networks. Concurrently, VGG16 neural network with a small number of parameters and a short time for additional training has proven to be highly efficient which is why it is recommended for the purposes of ship pattern recognition.

Keywords: artificial neural networks, pre-trained networks, convolutional neural networks, Keras, TensorFlow, Google Colaboratory, VGG16, NASNetMobile, NASNetLarge.

For citation: Konarev D.I., Gulamov A.A. Synthesis of neural network architecture for ship pattern recognition based on pre-training technology. *Modeling, Optimization and Information Technology*. 2022;10(2). Available from: https://moitvivt.ru/ru/journal/pdf?id=1148 DOI: 10.26102/2310-6018/2022.37.2.011 (In Russ.).

Введение

Согласно распоряжению Правительства РФ от 22.11.2008 N 1734-р при переходе к инновационному варианту развития транспортной системы необходимо обеспечить среди прочего расширение номенклатуры и повышение качества транспортных услуг на основе применения современных транспортных, логистических и информационно-телекоммуникационных технологий, развитие новых форм организации транспортного процесса и взаимодействия между видами транспорта. Указанная стратегия развития обусловливает актуальность создания системы инфокоммуникационного обеспечения судоходства.

Исходя из цели разработки методов и системы программно-технических средств сбора, обработки, хранения, анализа параметров судоходства, обеспечения навигации грузо- и пассажироперевозок в рамках информационно-телекоммуникационной системы мониторинга и управления судоходством по водным артериям основной задачей является разработка комплекса клиентских программных средств сбора и анализа параметров судоходства [1, 2, 3].

В области инфокоммуникационного обеспечения мониторинга судоходства основной подзадачей является непосредственно распознавание судов на изображениях. Эффективность применения нейронных сетей для решения этой задачи была показана в работе [4], в которой удалось синтезировать и обучить нейронную сеть, показывающую верные распознавания объектов в 82,12 % случаев.

Однако, обучение большого числа глубоких нейронных сетей занимает много времени. Возможно использование ускорителей вычислений (GPU) для сокращения времени обучения [5]. Но даже использование GPU может занять часы, дни и даже

месяцы для синтеза качественной глубокой нейронной сети с высокой точностью распознавания.

Другой вариант – использование готовой нейронной сети, которая уже обучена. Такие сети называются предварительно обученными сетями [6]. Предварительно обученная сеть уже способна решать конкретную задачу, а сведения об архитектуре такой сети и ее веса сохранены. Эти данные об уже обученной сети можно загрузить в программу и использовать для решения поставленных задач не тратя время на полное обучение. Так, в статье [7] для распознавания морских судов используется предварительно обученная нейронная сеть AlexNet. На данный момент это устаревшая сеть с низкими показателями качества распознавания. Подобная работа проделана и в статье [8] также с использованием переноса обучения. В ней используются более современные архитектуры, такие как Inception и ResNet. Таким образом, использование предварительно обученных нейронных сетей способно повысить эффективность инфокоммуникационного обеспечения судоходных путей и является актуальной задачей. В практической реализации это позволит с большей точностью определять параметры движения судов, таких как время прохода и скорость движения в точках расположения контролирующих камер для повышения контроля безопасности на водном транспорте.

Целью работы является исследование показателей эффективности распознавания судов наиболее современными и предварительно обученными сетями после их дообучения под поставленные задачи и выбор наиболее эффективной сети.

Материалы и методы

С развитием машинного обучения в последнее время все более удобным и популярным становится использование предварительно обученных нейронных сетей. Так как распознавание отдельных признаков на изображении является общей задачей, за которую обычно отвечает большая часть слоев нейронной сети, то данный метод заключается в дообучении под поставленную задачу лишь небольшой части нейронной сети. Это позволяет использовать очень глубокие нейронные сети, работающие с высокой точностью, при достаточно малых временных затратах. Как правило, дообучается лишь слой классификации. Материалом для дообучения является заранее подготовленный DataSet, который удовлетворяет требованиям поставленной задачи.

В Keras предварительно обученные сети содержатся в модуле Keras Applications. Модуль Keras Applications содержит несколько очень популярных предварительно обученных нейронных сетей, большая часть из которых используются для распознавания объектов на изображении [9]. Сеть VGG16 создана группой Visual Geometry Group из Оксфорда для распознавания объектов на изображении. Это достаточно глубокая сеть, которая содержит 16 слоев.

Сеть VGG19 – сеть той же самой группы из Оксфорда для распознавания объектов на изображениях, но более глубокая и содержит 19 слоев. Точность работы такой сети немного выше, чем в VGG16, но работает она медленнее из-за числа слоев.

Inception – сеть, которую разработала компания Google. Сеть для распознавания объектов на изображениях от компании Microsoft носит название ResNet50. В этой сети используется так называемое остаточное обучение (residual learning) [10]. Сеть имеет в соответствии с названием 50 слоев.

Xception – еще одна сеть для распознавания объектов на изображениях. Это модификация сети Inception от компании Google, от создателя Keras Франсуа Шолле.

Это самые популярные предварительно обученные нейронные сети для распознавания образов на изображениях. В модуле Keras Applications доступно большое число предварительно обученных сетей.

Все вышеперечисленные сети для распознавания объектов обучались на наборе данных ImageNet. Это открытый набор данных, который доступен на официальном сайте image-net.org и содержит 14 миллионов изображений [11]. Причем для каждого изображения есть метка с информацией о том, какой объект находится на этом изображении. Имеется большое количество объектов различных классов: люди, животные, растения, транспорт и т. д [12]. В наборе данных ImageNet классы объектов образуют иерархию, изображения, относящиеся к одному классу, разделены на категории, а те – на подкатегории [13]. Поэтому задачи исследования следующие.

1. Провести распознавание судов на изображениях с использованием ряда предварительно обученных сетей, дообучив их распознавать именно речные суда.

2. Провести выбор наиболее эффективной сети для данной задачи.

Первая сеть, которая была дообучена – VGG16. Это простая сеть, при этом качество ее работы достаточно высокое. Архитектура этой сети состоит из двух больших частей, первая часть, состоящая из каскадов свертки и подвыборки, отвечает за выделение характерных признаков изображения, а вторая часть, состоящая из трех полносвязанных слоев, отвечает за классификацию объектов на изображении на основании признаков, выделенных предыдущей сверточной частью сети [14]. Архитектура сети VGG16 представлена на Рисунке 1.

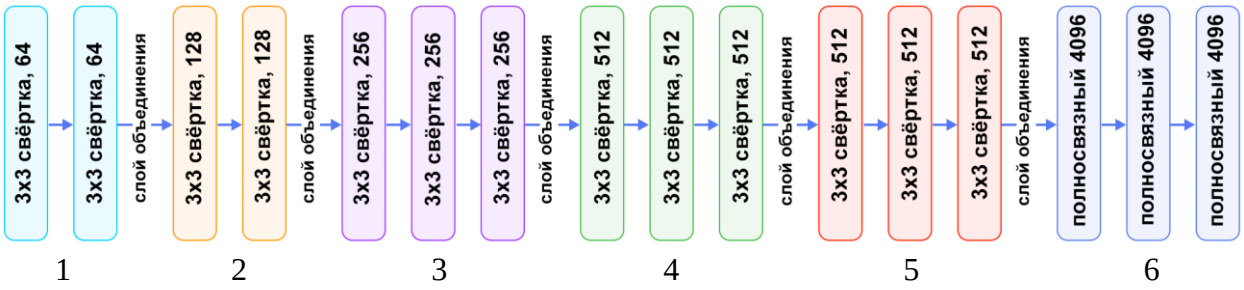

Рисунок 1 – Архитектура сети VGG16
Figure 1 – VGG16 network architecture

Сверточная часть состоит из пяти каскадов. Первые два каскада (1 и 2 блоки) включают по два уровня свертки с размером ядра свертки 3x3 и уровень подвыборки. Здесь используется выбор максимального значения из квадрата 2x2. Затем идут три каскада, в которых используются по три уровня свертки (3, 4 и 5 блоки) и уровень подвыборки. Как и на предыдущих слоях, размер сверточного узла здесь 3x3.

Три последних слоя отвечают за классификацию (6 блок), это полносвязанные слои. На последнем слое тысяча нейронов, которая соответствует тысячи классов объектов из набора данных ImageNet. Для классификации используется формат One-Hot Encoding [15].

Иными словами, выходными данными является разреженная матрица, в которой количество столбцов соответствует количеству распознаваемых классов объектов (в данном случае тысяча). В такой матрице нулю равны все значения кроме одного, которое соответствует классу объекта на распознаваемом изображении.

Подавляющее большинство методов классификации и регрессии сформулированы в терминах евклидовых или метрических пространств, то есть подразумевают представление данных в виде вещественных векторов одинаковой размерности. В реальных данных, однако, не так редки категориальные признаки,

принимающие дискретные значения. Чтобы было возможно применять линейные модели на таких данных, используется метод One-Hot Encoding [16]. Предположим, что некоторый признак может принимать 10 разных значений. В этом случае One-Hot Encoding подразумевает создание 10 признаков, все из которых равны нулю за исключением одного. На позицию, соответствующую численному значению признака, помещается 1. По умолчанию One-Hot Encoder преобразует данные в разреженную матрицу, чтобы не расходовать память на хранение многочисленных нулей.

Рассмотрим, как использовать сеть VGG16 в Keras. Весь код будем исполнять в Google Colab – сервисе, который позволяет запускать Jupyter проекты с применением GPU и в котором уже установлены все необходимые для машинного обучения библиотеки. Первая инструкция указывает, что необходимо использовать версию библиотеки TensorFlow 2.0 и выше. Со 2 по 6 строки происходит подключение различных компонентов библиотеки Keras. Оставшиеся инструкции подключают библиотеки для работы с файлами и облаком.

Далее подключается заранее подготовленный для дообучения DataSet.

Модуль VGG16 содержит необходимую сеть. На первом этапе необходимо создать модель, у которой будет архитектура сети VGG16. При создании также указываем, что хотим загрузить предварительно обученные веса на наборе данных ImageNet. Загружается предварительно обученная модель и выставляется флаг, который позволит дообучить модель.

При создании сети указываем параметр include_top – False, который означает, что не будет загружена та часть сети, которая отвечает за классификацию. То есть будет загружена только сверточная часть сети VGG16, которая отвечает за выделение характерных признаков на изображении [17]. Указываем размер тензора входных изображений 150 на 150 на 3. Так как необходимо сохранять минимальное число параметров сети, но при этом работать с такими размерами изображений, при которых остается возможным выделить характерные признаки, для большинства предварительно обученных нейронных сетей размер входного слоя равен 150 на 150. 3 канала стандартно используются для цвета. Также нужно указать, что сверточная часть в процессе обучения участвовать не будет, поэтому поле trainable сети устанавливаем в значении False. Загруженная сеть уже обучена выделять характерные признаки из изображений на наборе данных ImageNet, в который входят фотографии кораблей, то есть сеть уже умеет выделять нужные признаки, что существенно экономит время на обучении. Далее команда summary() выводит информацию о загруженной нейронной сети. Весь код на языке Python выполняется на площадке Google Colab, может быть воспроизведен и будет необходим и достаточен для воспроизведения рассмотренного в данной работе метода дообучения нейронных сетей. Google Colab – это бесплатный облачный сервис на основе Jupyter Notebook, который предоставляет интерфейс взаимодействия непосредственно в браузере и аппаратное ускорение вычислений на базе GPU и TPU. Однако данный код так же можно запускать и на локальной машине с использованием технологии CUDA, что потребует дополнительных настроек.

Так, предварительно обученная сеть VGG16 содержит 14 миллионов параметров, которые не требуют обучения.

На втором этапе создается составная сеть, которая включает скаченную сеть VGG16 и новую часть для классификации. То есть при создании такой сети в последовательную модель вместо отдельного слоя мы добавляем целую предварительно обученную сеть. Далее добавляются новые слои, которые будут использоваться для классификации. Первый слой типа Flatten – выпрямление. Этот слой нужен из-за того, что на выходе из сети VGG16 есть большое количество двумерных карт признаков, но на входе в полносвязный слой необходимо получить плоский вектор.

Таким образом, Flatten преобразует все эти двумерные карты признаков в одномерный вид. Затем идет полносвязанный слой, в котором 256 нейронов с полулинейной функцией активации. Слой Dropout для регуляризации снижения переобучения, выходной полносвязанный слой, в котором один нейрон и сигнальная функция активации. Эта функция плавно меняет свое значение от нуля до единицы, которая хорошо подходит для решения задачи бинарной классификации [18].

Первый слой – это отдельная модель VGG16, у которой 14 миллионов параметров. Затем идут те слои, которые были добавлены в модель, причем количество параметров для обучения – это веса и тех нейронов, которые входят в полносвязную часть сети. Параметры сверточной нейронной сети не обучаются.

После создания составной сети ее необходимо скомпилировать. Так как в качестве части составной сети использована сеть VGG16, которая уже обучена достаточно хорошо, необходимо использовать низкую скорость обучения [19]. При создании оптимизатора задаем параметр lr (сокращение от learning rate) равный единице на 10 в -5 степени. Если использовать более высокую скорость обучения, то алгоритм обучения не сойдется [20].

На третьем этапе необходимо загрузить изображения объектов, которые нужно распознавать. Прежде всего нужно подготовить обучающую, тестовую и проверочную выборки, то есть так называемый DataSet. Есть множество платформ, на которых размещаются общедоступные DataSet. Это такие сайты, как ImageNet, Analytics Vidhya и многие другие. Определение речных судов на изображении – задача бинарной классификации, для которой нужны положительные примеры (изображения с судами) и отрицательные примеры (изображения, на которых судов нет). Найти готовый DataSet на популярных площадках не удалось, поэтому был подготовлен свой DataSet. Для положительных примеров был использован DataSet с сайта Kaggle от индийского сообщества специалистов в области данных Analytics Vidhya [21]. DataSet включает в себя 6252 изображения грузовых и военных кораблей, кораблей-носителей, круизных лайнеров и танкеров. Примеры изображений представлены на Рисунке 2.

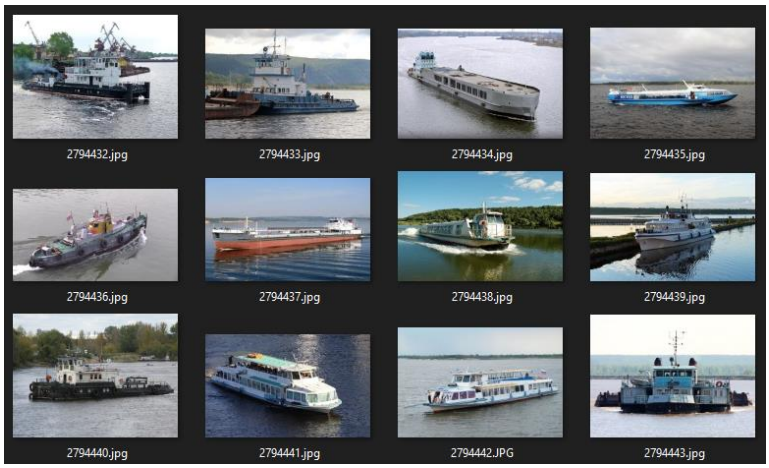

Рисунок 2 - Примеры изображений кораблей
Figure 2 - Examples of images with ships

Так как сверточные нейронные сети обучаются выделять характерные признаки заданного класса объектов, то наличие на фоне изображения корабля водной глади, неба, лесополосы или города не является существенным. Не менее важным является наличие в обучающем DataSet примеров с различных ракурсов, так как предполагается использование в системе множества камер, наблюдающих за судоходным фарватером

под разными углами. Это обеспечит возможность выделять нейронной сетью общие признаки судов вне зависимости от ракурса и фона.

Отрицательные примеры включают в себя любые изображения, на которых нет кораблей. Такой DataSet можно найти на сайте cocodataset.org [22]. DataSet включает в себя 5000 разнообразных размеченных изображений. Примеры изображений представлены на Рисунке 3.

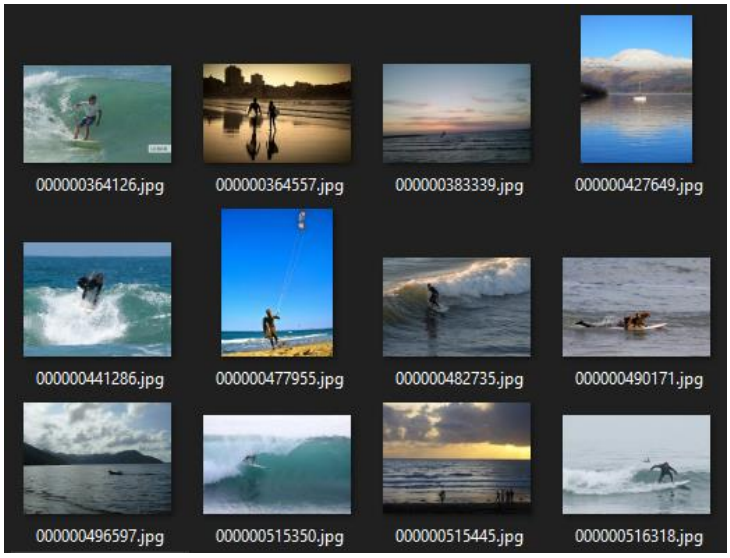

Рисунок 3 - Отрицательные примеры изображений
Figure 3 - Negative sample images

Из данных DataSet-ов был собран DataSet для дообучения предварительно обученных сетей под поставленную задачу. DataSet состоит из трех папок, представляющих обучающую, тестовую и проверочную выборки. В каждой из трех папок находятся еще по 2 папки, которые представляют классы изображений: изображения с речными судами и все остальные изображения. Это необходимо для модуля генератора изображений, который используется при обучении нейронных сетей.

На четвертом этапе следует подключить Google Drive к виртуальной машине.

Keras содержит средства для автоматической загрузки изображений и преобразования их в вид, пригодный для обучения нейронной сети с помощью так называемых генераторов. Для генераторов используется класс Keras ImageDatagenerator, который в данном случае использует всего один тип преобразования – деление каждого элемента изображения на 255. Таким образом, входными данными для нейронной сети являются изображения, преобразованные к разрешению 150 на 150 пикселей, а все элементы изображения будут находиться в диапазоне от нуля до единицы, и упакованы в массив значений, что удобно для обучения нейронной сети [23].

Затем используем метод flow_from_directory, который создает поток данных из каталога. Необходимо указать каталог, размер изображения, размер выборки, то есть количество изображений, которые будут прочитаны за один раз. Режим классов (class_mode) указывается как бинарный тип, так как имеется всего два класса, изображение речного судна и любое другое изображение. То есть на выходе имеется один нейрон, который выдает значение от 0 до единицы, соответствующее двум классам. В противном случае необходимо использовать тип «categorical».

Keras автоматически с помощью генератора загрузит фотографии из нужного каталога и сформирует правильные ответы для обучения с учителем. Аналогично создаются два других набора данных, набор данных для проверки, который будет

использоваться во время обучения для оценки гиперпараметров обучения, и набор данных для тестирования, который будет использоваться после окончания обучения для проверки качества работы сети на тех данных, которые не использовались во время обучения.

Далее обучается сеть с помощью генераторов методом обратного распространения ошибки. Так как используется предварительно обученная сеть, число эпох дообучения устанавливается небольшим, в данном случае 10. Для нейронных сетей, создаваемых с нуля, следует значительно повысить этот параметр.

После окончания обучения необходимо проверить качество работы сети. Для этого также используется генератор для тестовой выборки.

Результаты

Таким образом, разработана программа для создания составной сети на базе предварительно обученной сети и дообучения новой сети под поставленные задачи. А также произведена оценка точности дообученной сети с использованием тестовой выборки. Под точностью работы нейронной сети понимается отношение числа верно распознанных изображений к числу всех распознаваний. Точность работы дообученной нейронной сети VGG16 составила 94.38 %. Это значительно выше точности работы нейронной сети, синтезированной в работе [4], в которой оптимальный вариант архитектуры показал 82.12 % точности.

Аналогичным образом были дообучены все предварительно обученные сети, доступные в модуле Applications библиотеки Keras.

Результаты дообучения предварительно обученных сетей, доступных в модуле Applications библиотеки Keras приведены в Таблице 1. Результаты отсортированы по возрастанию качества сети.

Таблица 1 – Результаты дообучения предварительно обученных сетей, доступных в модуле Keras Applications
Table 1 – Results of pre-trained network retraining available in Keras Applications module

Название сети	Число параметров в сверточных слоях	Точность (%)
ResNet101	13,107,713	64.06
ResNet152	13,107,713	64.06
ResNet50	13,107,713	64.11
DenseNet121	7,037,504	78.08
MobileNet	3,228,864	76.90
MobileNetV2	2,257,984	76.95
InceptionResNetV2	54,336,736	81.20
DenseNet201	18,321,984	81.98
DenseNet169	12,642,880	83.06
ResNet152V2	13,107,713	85.84
InceptionV3	21,802,784	91.80
Xception	20,861,480	91.85
VGG19	20,024,384	92.97

Таблица 1 (продолжение)
Table 1 (continued)

Название сети	Число параметров в сверточных слоях	Точность (%)
VGG16	14,714,688	94.38
NASNetMobile	84,916,818	98.49
NASNetLarge	84,916,818	98.97

Обсуждение

Как видно из таблицы, наибольшей точностью обладают сети NASNetMobile и NASNetLarge. Но их использование потребовало изменений программы, в отличие от всех остальных сетей, которые работают с изображениями 150 на 150 пикселей, в связи с тем, что минимально допустимый размер изображений для работы с сетями NASNetMobile и NASNetLarge – 331 на 331 пиксель. Данное различие обусловлено архитектурой сетей NASNetMobile и NASNetLarge и не является существенным, так как целью исследования является поиск наиболее точной и быстрой модели. Также, ввиду большого числа параметров и размера изображений, время дообучения данных сетей оказалось намного больше в сравнении с остальными сетями и заняло несколько часов. Быстродействие таких сетей также намного ниже.

Выводы

Исходя из полученных результатов, рекомендуется использовать сеть VGG16, так как при малом числе параметров и высоком быстродействии она показывает очень высокую эффективность. Так, при сравнимой точности работы VGG16, NASNetMobile и NASNetLarge (94.38 %, 98.49 % и 98.97 % соответственно), их дообучение заняло 418 секунд для 10 эпох, 2146 секунд для 10 эпох и 4391 секунду для 5 эпох соответственно. В среднем одна эпоха обучения VGG16 заняла 41.7 секунды, одна эпоха обучения NASNetMobile заняла 3 минуты 34.6 секунды, одна эпоха обучения NASNetLarge заняла 14 минут 38.2 секунды.

Так как в публичном доступе для поставленной задачи нет подходящего набора данных, был создан DataSet для обучения нейронных сетей распознаванию образов речных судов на изображениях. Также проведен ряд экспериментов и на базе современных предварительно обученных нейронных сетей путем дообучения. Были получены сети, решающие поставленную задачу с высокой точностью и определены наиболее эффективные.

СПИСОК ИСТОЧНИКОВ

1. Гольцова И.А., Гуламов А.А. Информационное обеспечение участка железной дороги *Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение*. 2017;7(2):6–11. Доступно по: https://swsu.ru/izvestiya/seriesivt/archiv/2_2017.pdf.
2. Маклаков Е.С., Гуламов А.А. Узел сбора информации диспетчерского центра *Известия Юго-Западного государственного университета*. 2018;22(6):136–142. Доступно по: https://doi.org/10.21869/2223-1560-2018-22-6-136-142.
3. Маклаков Е.С., Гуламов А.А. Оптимизация «последних миль» до удаленных узлов доступа путем применения технологии LCAS *Моделирование, оптимизация и*

информационные технологии. 2019;7(3). Доступно по: https://moitvivt.ru/ru/journal/pdf?id=635. DOI: 10.26102/2310-6018/2019.26.3.039.
4. Гуламов А.А., Конарев Д.И. Синтез архитектуры нейронной сети для распознавания образов морских судов. *Известия Юго-Западного государственного университета*. 2020;24(1):130–143. Доступно по: https://doi.org/10.21869/2223-1560-2020-24-1-130-143.
5. Bastiaan Sjardin, Luca Massaron, Alberto Boschetti. *Large Scale Machine Learning with Python.* Packt Publishing; 2016. 420 p.
6. Adrian Rosebrock. *Deep Learning for Computer Vision with Python*. PyImageSearch, 2017. 330 p.
7. Cuong Dao-Duc, Hua Xiaohui, Olivier Morère. Maritime Vessel Images Classification Using Deep Convolutional Neural Networks. *SoICT*. 2015:276–281. Доступно по: https://doi.org/10.1145/2833258.2833266.
8. Leclerc M., Tharmarasa R., Florea M.C., Boury-Brisset A.C., Kirubarajan T., Duclos-Hindie N., Ship classification using deep learning techniques for maritime target tracking. *21 st International Conference on Information Fusion FUSION*. 2018:737–744 pp.
9. Tom Hope, Yehezkel S. Resheff. *Itay Lieder Learning TensorFlow: A Guide to Building Deep Learning Systems*. O'Reilly Media; 1 edition; 2017. 242 pp.
10. Мюллер А. *Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными*. Вильямс; 2017. 480 с.
11. Себастьян Рашка. *Python и машинное обучение*. ДМК-Пресс; 2017. 418 с.
12. Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, *Google Brain* Tensor flow: A system for large-scale machine learning. *Operating Systems Design and Implementation: Proc. 12th Symposium*, Savannah, GA, USA. 2016:265–283.
13. Антонио Джулли, Суджит Пал. *Библиотека Keras – инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и Tensor Flow*. ДМК-Пресс; 2017. 296 с.
14. Франсуа Шолле. *Глубокое обучение на Python*. Питер; 2018. 400 с.
15. Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media; 2017. 574 p.
16. Ian Goodfellow. *Deep Learning (Adaptive Computation and Machine Learning series)*. The MIT Press; 2016. 800 p.
17. Tariq Rashid. *Make Your Own Neural Network*. CreateSpace Independent Publishing Platform; 2016. 222 p.
18. Schmidhuber J. Deep learning in neural networks: An overview. *Neural Networks*. 2015;(61):85–117.
19. Josh Patterson, Adam Gibson. *Deep Learning: A Practitioner's Approach*.; 2017. 532 p.
20. Саймон Хайкин. *Нейронные сети*. Вильямс; 2018. 1104 с.
21. *Kaggle: Your Machine Learning and Data Science Community Game of Deep Learning: Ship datasets*. 2019. Доступно по: https://www.kaggle.com/arpitjain007/game-of-deep-learning-ship-datasets (дата обращения: 10.03.2021).
22. *COCO: Common Objects in Context 2017 Val images.* 2017. Доступно по: http://images.cocodataset.org/zips/val2017.zip (дата обращения: 10.03.2021).
23. Michael Taylor. *The Math of Neural Networks*. Amazon Digital Services LLC – Kdp Print Us; 2017. 168 p.

REFERENCES

1. Gol'cova I.A., Gulamov A.A. Informacionnoe obespechenie uchastka zheleznoj do-rogi *Izvestija Jugo-Zapadnogo gosudarstvennogo universiteta. Serija: Upravlenie, vychislitel'naja tehnika, informatika. Medicinskoe priborostroenie*. 2017;7(2):6–11. Available at: https://swsu.ru/izvestiya/seriesivt/archiv/2_2017.pdf. (In Russ.)
2. Maklakov E.S., Gulamov A.A. Uzel sbora informacii dispetcherskogo centra. *Izvestija Jugo-Zapadnogo gosudarstvennogo universiteta*. 2018;22(6):136–142. Available at: https://doi.org/10.21869/2223-1560-2018-22-6-136-142. (In Russ.)
3. Maklakov E.S., Gulamov A.A. Optimizacija «poslednih mil'» do udalennyh uzlov dostupa putem primenenija tehnologii LCAS *Modelirovanie, optimizacija i informacionnye tehnologii = Modeling, Optimization and Information Technology*. 2019;7(3). Available at: https://moitvivt.ru/ru/journal/pdf?id=635. DOI: 10.26102/2310-6018/2019.26.3.039. (In Russ.)
4. Gulamov A.A., Konarev D.I., Cintez arhitektury nejronnoj seti dlja raspoznavanija obrazov morskih sudov. *Izvestija Jugo-Zapadnogo gosudarstvennogo universiteta*. 2020;24(1):130–143. Available at: https://doi.org/10.21869/2223-1560-2020-24-1-130-143. (In Russ.)
5. Bastiaan Sjardin, Luca Massaron, Alberto Boschetti. *Large Scale Machine Learning with Python*, Packt Publishing; 2016. 420 p.
6. Adrian Rosebrock. *Deep Learning for Computer Vision with Python*. PyImageSearch; 2017. 330 p.
7. Cuong Dao-Duc, Hua Xiaohui, Olivier Morère. Maritime Vessel Images Classification Using Deep Convolutional Neural Networks. *SoICT*. 2015:276–281. Available at: https://doi.org/10.1145/2833258.2833266.
8. Leclerc M., Tharmarasa R., Florea M.C., Boury-Brisset A.C., Kirubarajan T., Duclos-Hindie N., Ship classification using deep learning techniques for maritime target tracking, *21 st International Conference on Information Fusion FUSION*. 2018:737–744.
9. Tom Hope, Yehezkel S. Resheff. *Itay Lieder Learning TensorFlow: A Guide to Building Deep Learning Systems*. O'Reilly Media; 1 edition; 2017. 242 pp.
10. Andreas Mjuller. *Vvedenie v mashinnoe obuchenie s pomoshh'ju Python. Rukovodstvo dlja specialistov po rabote s dannymi*. Vil'jams; 2017. 480 s. (In Russ.)
11. Sebast'jan Rashka. *Python i mashinnoe obuchenie*. DMK-Press; 2017. 418 s. (In Russ.)
12. Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, *Google Brain* Tensor flow: A system for large-scale machine learning. *Operating Systems Design and Implementation: Proc. 12th Symposium*, Savannah, GA, USA, 2016: 265–283.
13. Antonio Dzhulli, Sudzhit Pal. *Biblioteka Keras - instrument glubokogo obuche-nija. Realizacija nejronnyh setej s pomoshh'ju bibliotek Theano i Tensor Flow*. DMK-Press; 2017. 296 p. (In Russ.)
14. Fransua Sholle. *Glubokoe obuchenie na Python*. Piter; 2018. 400 p. (In Russ.)
15. Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media; 2017. 574 p.
16. Ian Goodfellow. *Deep Learning (Adaptive Computation and Machine Learning series)*. The MIT Press; 2016. 800 p.
17. Tariq Rashid. *Make Your Own Neural Network*. CreateSpace Independent Publishing Platform; 2016. 222 p.

18. Schmidhuber J. Deep learning in neural networks: An overview. *Neural Networks*. 2015;(61):85–117.
19. Josh Patterson, Adam Gibson. *Deep Learning: A Practitioner's Approach*.; 2017. 532 p.
20. Sajmon Hajkin. *Nejronnye seti.* Vil'jams; 2018. 1104 p. (In Russ.)
21. *Kaggle: Your Machine Learning and Data Science Community Game of Deep Learning: Ship datasets*. 2019. Available at: https://www.kaggle.com/arpitjain007/game-of-deep-learning-ship-datasets (accessed on: 10.03.2021).
22. *COCO: Common Objects in Context 2017 Val images*. 2017. Available at: http://images.cocodataset.org/zips/val2017.zip (accessed on: 10.03.2021).
23. Michael Taylor. *The Math of Neural Networks*. Amazon Digital Services LLC –Kdp Print Us; 2017. 168 p.

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Дмитрий Игоревич Конарев, аспирант, кафедра космического приборостроения и систем связи Юго-Западного государственного университета, Курск, Российская Федерация.
e-mail: dmitrii.konarev@gmail.com
ORCID: 0000-0002-1161-4767

Dmitry Igorevich Konarev, Postgraduate Student of the Department of Space Instrumentation and Communication Systems of Southwestern State University, Kursk, Russian Federation.

Алишер Абдумаликович Гуламов, доктор физико-математических наук, профессор кафедры космического приборостроения и систем связи Юго-Западного государственного университета, Курск, Российская Федерация.
e-mail: profgulamov@mail.ru
ORCID: 0000-0001-9334-6710

Alisher Abdumalikovich Gulamov, Doctor of Physical and Mathematical Sciences, Professor of the Department of Space Instrumentation and Communication Systems of Southwestern State University, Kursk, Russian Federation.

Статья поступила в редакцию 14.03.2022; одобрена после рецензирования 04.05.2022; принята к публикации 23.05.2022.

The article was submitted 14.03.2022; approved after reviewing 04.05.2022; accepted for publication 23.05.2022.