



научный электронный журнал
ПРИНЦИПЫ ЭКОЛОГИИ

<http://ecopri.ru>

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ



<http://petrsu.ru>

Издатель

ФГБОУ ВО «Петрозаводский государственный университет»
Российская Федерация, г.Петрозаводск, пр.Ленина,33

Научный электронный журнал

ПРИНЦИПЫ ЭКОЛОГИИ

<http://ecopri.ru>

№ 4 (58). Декабрь, 2025

Главный редактор

А. В. Коросов

Редакционный совет

В. Н. Большаков
А. В. Воронин
Н. Н. Немова
Г. С. Розенберг
А. Ф. Титов
Г. С. Антипина
В. В. Вапиров
А. М. Макаров

Редакционная коллегия

Т. О. Волкова
Е. П. Иешко
В. А. Илюха
Н. М. Калинкина
J. P. Kurhinen
А. Ю. Мейгал
J. B. Jakovlev
B. Krasnov
A. Gugolek
В. Н. Якимов
А. В. Сони́на

Службы поддержки

А. Г. Марахтанов
Е. В. Голубев
С. Л. Смирнова
Н. Д. Чернышева
М. Л. Киреева

ISSN 2304-6465

Адрес редакции

185910, Республика Карелия, г.Петрозаводск, пр. Ленина, 33. Каб. 453

E-mail: ecopri@psu.karelia.ru

<http://ecopri.ru>



© ФГБОУ ВО «Петрозаводский государственный университет»



УДК 57.087:528.854

ИСПОЛЬЗОВАНИЕ БИБЛИОТЕКИ KERAS ДЛЯ ДЕШИФРИРОВАНИЯ МЕСТООБИТА- НИЙ ЖИВОТНЫХ МЕТОДАМИ ГЛУБОКОГО ОБУЧЕНИЯ

МАРФИЦЫНА
Наталья Александровна

*ФГБОУ ВО Петрозаводский государственный университет,
185910, Россия, Республика Карелия, г. Петрозаводск, пр. Лени-
на, 33, marfitsyna.nata@mail.ru*

КОРОСОВ
Андрей Викторович

*доктор биологических наук, Петрозаводский государственный
университет, Петрозаводск, пр. Ленина, 33,
korosov@psu.karelia.ru*

Ключевые слова:

местообитания
ДЗ
ГИС
нейронная сеть
R
Keras

Аннотация: В работе рассматривается применение алгоритмов глубокого обучения из библиотеки Keras для решения задачи классификации вырубок разного возраста с помощью дистанционного зондирования в среде R. Подробно рассмотрена достаточно сложная процедура установки библиотек Keras на компьютер. Описаны этапы нейросетевого моделирования и их вариации при использовании пакета R neuralnet и среды Keras. Выполнено дешифрирование космических снимков в окрестностях д. Гомсельга (Карелия) с использованием данных полевой съемки. Типичный алгоритм дешифрирования (классификация с обучением) был дополнен совместным многомерным анализом яркостных характеристик снимка и полевыми геоботаническими описаниями. В результате были сформированы четыре набора эталонных сигнатур, соответствующие тому или иному состоянию зарастающих вырубок. Нейросеть (многослойный перцептрон) настраивалась на распознавании этих типов насаждений и затем выполнила классификацию остальных пикселей снимка для всей изучаемой территории. На основе анализа геоботанических описаний и спутниковых данных была создана грид-карта с выделением четырех основных типов местообитаний: свежие вырубки, зарастающие вырубки, молодняки, лиственный лес. Обработка данных с помощью алгоритмов Keras существенно ускоряет анализ, позволяет увеличивать число слоев и нейронов и детализировать грид. В частности, в отличие от алгоритмов эталонного дешифрирования, предлагаемый подход позволил выявить неоднородность растительности в пределах одновозрастных вырубок. Результаты работы используются для выявления разнородных местообитаний животных и влияния экологических факторов на их пространственное распределение и численность.

© Петрозаводский государственный университет

Рецензент: И. Р. Идрисов
А. Н. Шихов

Получена: 16 октября 2025 года

Введение

Биотопическая гетерогенность является ключевым фактором, определяющим распределение и динамику численности наземных позвоночных животных. В таежных эко-

Подписана к печати: 27 декабря 2025 года

системах Карелии антропогенные нарушения, такие как сплошные рубки, инициируют сукцессионные процессы, формирующие мозаику вторичных лесов с уникальными микроклиматическими и кормовыми усло-

виями. Эти изменения оказывают прямое воздействие на структуру популяций мелких млекопитающих и их паразитических сообществ, что подтверждено многолетними исследованиями (Коросов и др., 2003; Иешко и др., 2020). Ранние наши работы были направлены на описание трансформации местообитаний с применением методов дистанционного зондирования (ДЗ) и геоинформационного анализа (Бугмырин и др., 2006; Гусева и др., 2014). Современные технологии, а именно нейросетевое моделирование, позволяют по-новому подойти к этому процессу и построить карту различающихся биотопов, отражающих роль разнообразных экологических факторов в формировании уровня и динамики численности изучаемых групп животных. Однако использование космических снимков существенно увеличивает объем входящих данных, обработка которых в среде интерпретатора R затруднена и требует много времени. Эту проблему решает библиотека глубокого обучения Keras (Шолле, 2022), позволяющая компилировать скрипты. Кроме того, в среде Keras предусмотрено использование других видов нейронных сетей, кроме персептрона.

Цель работы состоит в демонстрации построения нейросетевой модели с использованием пакета Keras на примере решения экологической задачи – классификации разновозрастных молодых вторичных насаждений на месте рубок дистанционными методами на основе натурных геоботанических описаний.

Статья носит методический характер, и любой читатель может воспроизвести расчеты с помощью представленных скриптов и наших данных, которые доступны по гиперссылкам.

Материалы

Полевые данные получены в 2023 г. в окрестностях д. Малая Гомсельга (Кондопожский район, Карелия, N 62.067, E 33.9587). Проводились геоботанические описания по упрощенной схеме только во вторичных лесах на вырубках в возрасте 2–30 лет. Фиксировались 16 характеристик: сомкнутость крон (trans, доля белых пикселей бинарных снимков при съемке вертикально вверх с точки учета, %), полнота древостоя с помощью полнотомера Биттерлиха (с – сосна, е – ель, б – береза, ос – осина, ол – ольха, и – ива, г – рябина, шт.), число поваленных деревьев в окрестностях 20 м от точки (fell, шт.), проективное покрытие в окрестностях 5 м от точки

учета (Mxi – мхи, Lish – лишайники, Travi – травы, kust – кустарники, Pap – папоротники, Mal – малина, %). Получено 120 описаний на 40 площадках; все точки имеют картографическую привязку. Описания на отдельной площадке (в одном биотопе) выполнялись в трех точках, отстоящих друг от друга на 10 м. В результате была получена таблица размером 16 полей и 120 строк. Далее для каждой площадки три описания обобщались: для каждого показателя была рассчитана медиана и стандартное отклонение, всего 32 показателя для отдельной площадки. В заключение для каждого показателя были высчитаны нормированные отклонения, отношение разности между i -м показателем и средней к стандартному отклонению, $(x^*Mx)/Sx$, что приводит разную размерность переменных к безразмерному диапазону 0 ± 3 и сохраняет уровень коррелированности между переменными. Так получили рабочую матрицу, состоящую из 32 полей (характеристик биотопа) и 40 строк (площадки в биотопах).

Использованы космические снимки за март и июль 2023 г., из которых вырезали фрагменты по экстену: 546015, 6877965, 553785, 6887535 (WGS 84 / UTM zone 36N). Взяты 7 каналов из летнего снимка: [1_sum23.tif](#), [2_sum23.tif](#), [3_sum23.tif](#), [4_sum23.tif](#), [5_sum23.tif](#), [6_sum23.tif](#), [7_sum23.tif](#) и 2 канала из зимнего снимка: [1_win23.tif](#), [2_win23.tif](#). Разрешение снимков составило 30 м на пиксель. Для каждого канала проводили предобработку – нормализацию значений в пределах от 0 до 255. Полный набор данных по яркости всех пикселей для 9 каналов спутниковых снимков в пределах изученного экстенга поместили в массив [fuldata](#).

Далее были подготовлены массивы с яркостными характеристиками для выполнения анализа. Используя координаты точек описания, определили пиксели, покрывающие участки описания (120 участков для 40 площадок). Сформирована таблица [bancl.csv](#) размером 9 полей и 120 строк для обучения нейронных сетей. Яркостные характеристики этих пикселей по всем каналам снимков использовали для составления обучающих и тестовых массивов (тензоров). Данные были случайным образом разбиты на два массива (два тензора). Обучающий тензор, **train_data**, содержал 70 % данных (28 площадок). Тестовый тензор, **test_data**, включал оставшиеся 30 % (12 площадок).

Подготовку массивов данных ({terra}), кластерный анализ (hclust {stats}) и нейросетевое моделирование (neuralnet {neuralnet},

compile{keras}) выполнили в среде R (The R Project..., 2023).

Традиционные методы исследований

Традиционная схема дешифрирования с обучением использует данные полевых описаний и космические снимки (Геоинформатика, 2005). По имеющимся натурным данным намечаются эталонные участки с заведомо разными типами природных объектов (для которых типология уже известна). Тем самым пиксели разных областей снимка обретают определенное качество, а именно тип природного объекта. Далее выполняется тот или иной вариант классификации пикселей (с помощью дискриминантного, кластерного или нейросетевого анализа). Вначале отыскивается уравнение, которое связывает значения яркости пикселей (сигнатуры) с разными типами указанных природных объектов (этап обучения). Затем это уравнение используется для экстраполяции (прогноза) заданных классов объекта на изучаемую область космического снимка.

Иногда эталонные участки назначаются с использованием карты лесоустроительных выделов (Ильчук, Цай, 2010; Кузменко и др., 2015). В одних случаях она служит для построения прямоугольных участков в рамках выдела, которые применяются для считывания яркостных характеристик со спутниковых снимков, в других случаях в качестве маски используются некоторые выделы целиком в своих границах. Для автоматизации этого процесса в один блок могут объединяться данные векторной карты лесоустройства и яркостные характеристики спутниковых снимков, что позволяет вычислить статистические характеристики для каждого выдела на основе множества пикселей растровых слоев спутникового изображения, попадающих в границы выдела (Данилова и др., 2017).

В нашем случае типология объектов была изначально непонятна, и мы применили другой алгоритм для построения схемы классификации типов растительного покрова. Сначала в полевых условиях выполнили серию геоботанических описаний. Далее в одну матрицу объединили натурные полевые данные и яркостные характеристики пикселей снимка, покрывающих пробные участки. Затем выполняли классификацию (кластеризацию) объектов по объединенным данным, получив небольшое число классов, тем самым выявив типологию изучаемых объектов.

Процедура выделения типов биотопов (кластеров clu) подробно изложена в работе по дешифрированию местообитаний животных (Коросов, Марфицына, 2025). Она включает совместный кластерный анализ массива данных по 41 показателю (9 спектральных каналов и 32 геоботанические характеристики) для 40 площадок. Сначала с помощью алгоритма *k*-средних и «метода локтя» выявляется минимальное число «естественных» кластеров (Шитиков, Мاستицкий, 2017), затем процедура кластеризации методом Варда делит все множество площадок на принятое число кластеров. Результатом стали 4 интерпретированных кластера, соответствующих стадиям сукцессии: свежие вырубki (кластер 1 для 4 площадок), лиственный лес (кластер 2 для 9 площадок), молодняки (кластер 3 для 7 площадок), зарастающие вырубki (кластер 4 для 16 площадок). Кластерные метки (clu) площадок далее используются как целевые входные переменные для обучения нейронной сети на основе спектральных характеристик (9 каналов).

Поскольку каждому типу биотопа соответствуют определенные значения яркостей пикселей, выполнили обучение сети для прогноза типа объекта по растровым данным. Затем выполнили экстраполяцию полученных зависимостей на весь снимок. В результате получили цветной грид, где отдельные одноцветные пятна соответствуют тому или иному типу природных объектов (биотопов). Используя нейронную сеть, для каждого пикселя геоизображения изучаемой территории рассчитали свой номер кластера. Пятна, сформированные пикселями одного цвета, рассматривали как отдельные типы биотопов.

Описанная процедура в какой-то мере справлялась с поставленными задачами, если объем баз данных, площадь изучаемых территорий, размеры космических снимков и число нейронов сети относительно невелики. Для более точного прогноза нами принята большая серия попыток существенно увеличить число слоев и нейронов в персептроне (пакет *neuralnet*). При этом резко увеличивается время обучения сети и число сбоев в расчетах, причем без существенного роста качества полученной модели, которая зачастую оказывается переученной. Для таких задач (тем более с *bigdata*) средств пакетов *neuralnet* недостаточно и требуются более мощные библиотеки, такие как Keras.

Оригинальные методы исследований

Рассмотрим решение поставленной выше задачи с помощью библиотеки Keras. Установка Keras на компьютере требует кропотливой работы. В литературе и в интернете можно найти ряд разрозненных описаний, как это сделать (Шолле, 2022; PythonRu, 2021; GeeksforGeeks, 2021), ниже рассмотрен наш опыт по установке и использованию Keras. Библиотеку следует устанавливать на компьютеры не старше 8 лет, выпущенные после 2017 г. На старых устройствах могут возникнуть ошибки, связанные с невозможностью виртуальной среды найти совместимую версию TensorFlow для текущей версии Python. В таких случаях необходимо скачать дополнительные пакеты, такие как Rtools, которые требуются для сборки пакетов R. Также могут возникнуть проблемы с загрузкой динамической библиотеки TensorFlow. Библиотека имеет определенные системные требования, включая наличие Microsoft C++ Redistributable for Visual Studio 2015, 2017 и 2019. Важно убедиться в совместимости версий Python и TensorFlow, чтобы избежать проблем с установкой и работой системы.

Установка Keras

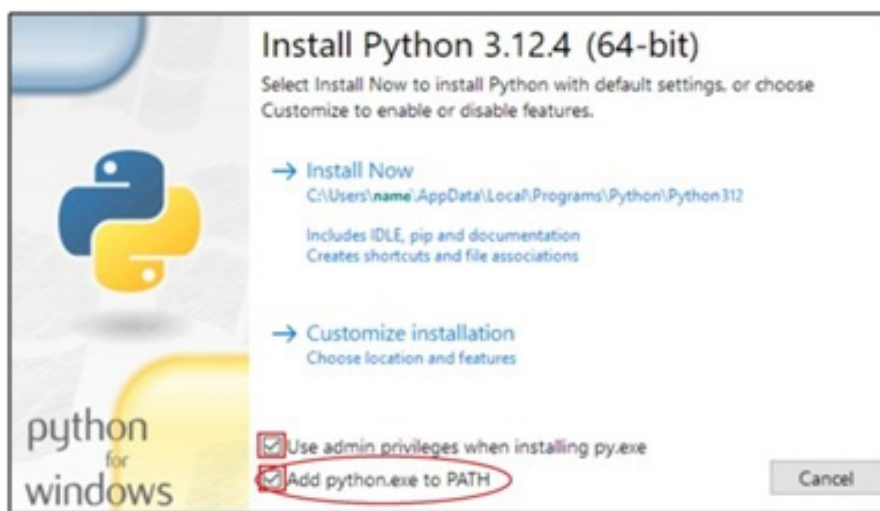
Библиотека Keras работает в относительно обособленной виртуальной среде, созданной в компьютере с помощью дополнительных программ. Мы использовали следующие программы: Python (на нем напи-

сан Keras), Rstudio (оболочка для работы со скриптами R), Tensorflow (библиотека программ), Numpy (библиотека программ). Во время установки этих программ необходимо пользоваться командной строкой Windows. Программы устанавливаются по определенному алгоритму в искомой виртуальной среде. Рекомендуется выполнять их загрузку от имени администратора.

Первый этап. Подготовка системы

Скачиваем загрузочные файлы и производим одновременно установку. Скачивание файлов производится с открытых официальных серверов, поскольку все программы условно бесплатны. Программа R должна быть установлена заранее.

Загружаем инсталлятор Python 3.12.4 с сайта <https://www.python.org/downloads/>. Важно: на сегодняшний день остальные библиотеки корректно устанавливаются только с версией Python не выше 3.12; современным читателям необходимо скачать не последнюю версию, а именно Python 3.12. Запускаем инсталлятор. В открывшемся окне установщика Install Python внизу необходимо отметить обе опции «Use admin privileges when installing py.exe» и «Add python.exe to PATH». Это позволит добавить путь к установленному Python в системную переменную PATH и завершить процесс установки программы. Нажимаем кнопку Install Now. В окне Setup was successful нажимаем кнопку Close.



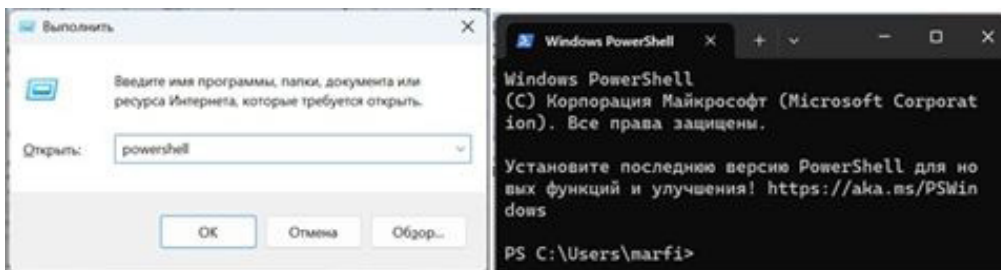
Загружаем инсталлятор Rstudio 2024.09.1 с сайта <https://posit.co/download/rstudio->

[desktop/](https://posit.co/download/rstudio-desktop/) и запускаем установку (оба процесса занимают много времени).

Второй этап. Настройка виртуальной среды

Вызываем окно Выполнить, нажав соче-

тание клавиш **Win + R**, вводим с клавиатуры команду **powershell**, ОК. Появляется окно командной строки.



Проверяем успешность установки Python. Для этого в командной строке вводим команду **python --version**, ждем клавишу Enter.

Если Python установлен корректно, появится строка: Python 3.12.4 (с версией программы).



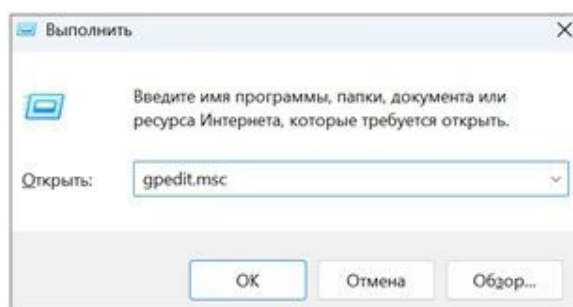
Проверяем наличие программы pip, которая устанавливается вместе с Python. В командной строке вводим команду **pip --version**, Enter. Должна появиться строка

с версией программы и путем до нее: pip 25.0.1 from C:\Python 312\Lib\site-packages\pip (python3.12).



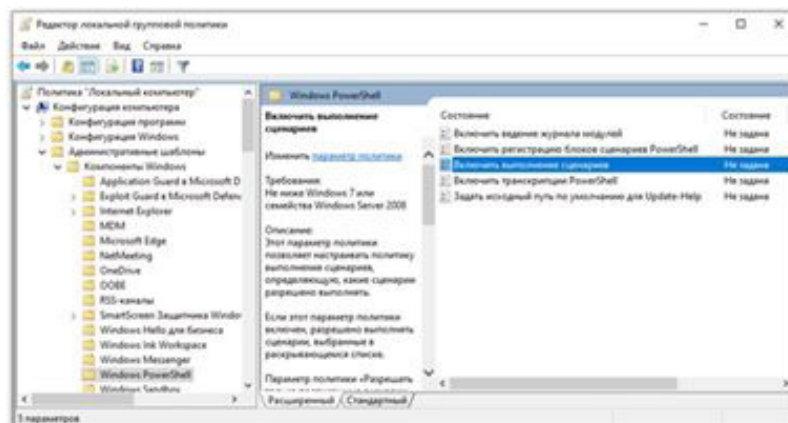
Даем разрешение на выполнение сценариев. Вновь вызываем окно Выполнить (Win

+ R) и вводим команду **gpedit.msc**, ОК.



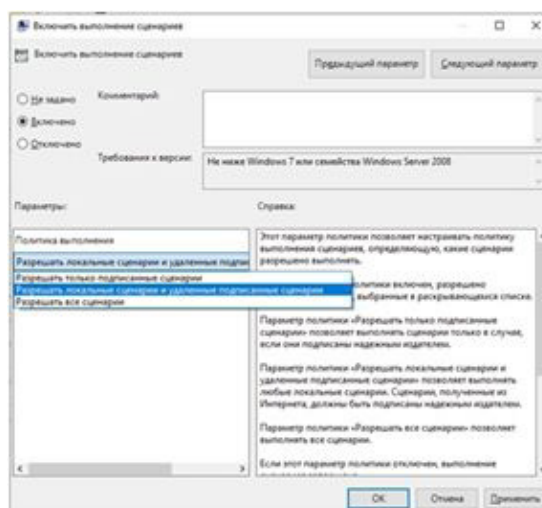
В появившемся окне Редактор локальной групповой политики необходимо *включить выполнение сценариев*. Для этого в левой части окна поочередно отрываем вкладки Конфигурация компьютера \ Администра-

тивные шаблоны \ Компоненты Windows \ Windows PowerShell и в правой части окна дважды кликаем на пункте Включить выполнение сценариев.



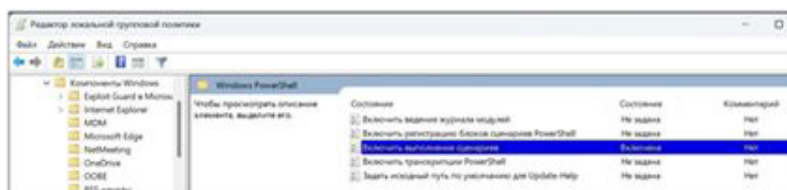
В появившемся окне Включить выполнение сценариев выбрать пункт Включено и, нажав на серую полосу в левой части окна,

выбрать Разрешать локальные сценарии и удаленные подписанные сценарии, ОК.



В появившемся окне Редактор... состоящие работы «Включить выполнение сценариев» изменится на «Включено». Закрываем окно.

риев» изменится на «Включено». Закрываем окно.



Третий этап. Создание виртуального окружения
Вновь вызываем командную строку (см.

выше: Win + R, powershell, ОК). Вводим команду **pip install virtualenv**, Enter.



По выполнении должна появиться строка

```
Installing collected packages: distlib, platformdirs, filelock, virtualenv
Successfully installed distlib-0.3.9 filelock-3.18.0 platformdirs-4.3.7 virtualenv-20.30.0
```

В следующей строке ввести команду **python -m venv r-reticulate**, Enter.

```
PS C:\Users\marfi> python -m venv r-reticulate
```

И, наконец, активировать окружение командой **r-reticulate\Scripts\activate**, Enter. Успешная активация будет обозначаться зеленой надписью «r-reticulate».

```
PS C:\Users\marfi> r-reticulate\Scripts\activate
(r-reticulate) PS C:\Users\marfi> |
```

Четвертый этап. Установка библиотек для установки необходимых библиотек: **pip install tensorflow keras matplotlib pandas numpy**.
В командной строке ввести команды

```
(r-reticulate) PS C:\Users\marfi> pip install tensorflow
keras matplotlib pandas numpy
```

Начнется процесс инсталляции. По завершении процесса должна быть выведена соответствующая текстовая строка.

```
Successfully installed MarkupSafe-3.0.2 astunparse-1.6.3 certifi-2025.1.31 charset-normalizer-3.4.1 flatbuffers-25.2.1
gast-0.6.0 google-pasta-0.2.0 grpcio-1.71.0 idna-3.10 libclang-18.1.1 markdown-3.8 numpy-2.1.3 opt-einsum-3.4.0 prot
obuf-5.29.4 requests-2.32.3 setuptools-79.0.0 tensorboard-2.19.0 tensorboard-data-server-0.7.2 tensorflow-2.19.0 termc
olor-3.0.1 urllib3-2.4.0 werkzeug-3.1.3 wheel-0.45.1 wrapt-1.17.2
```

Чтобы проверить наличие установленных библиотек, используем команды **pip show tensorflow** и **pip list**.

<pre>PS C:\Users\marfi> pip list Package Version ----- absl-py 2.1.0 astunparse 1.6.3 certifi 2024.7.4 charset-normalizer 3.3.2 flatbuffers 24.3.25 gast 0.6.0 google-pasta 0.2.0 grpcio 1.65.1 h5py 3.11.0 idna 3.7 keras 3.4.1</pre>	<pre>PS C:\Users\marfi> pip show tensorflow Name: tensorflow Version: 2.17.0 Summary: TensorFlow is an open source machine learning framework for everyone. Home-page: https://www.tensorflow.org/ Author: Google Inc. Author-email: packages@tensorflow.org License: Apache 2.0 Location: C:\Python 312\Lib\site-packages Requires: tensorflow-intel Required-by: PS C:\Users\marfi></pre>
--	---

Если все пройдет без ошибок, закроем окно Windows PowerShell.

Пятый этап. Построение виртуального окружения

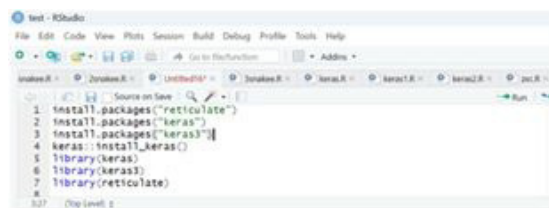
Запустим программу Rstudio, играющую роль оболочки для R, где создаются и выполняются скрипты. Для оптимизации работы периодически необходимо перезапускать сессию R (команда главного меню Session \ Restart R).

В открывшемся окне создадим скрипт командой File \ New File \ R-script.

Запишем серию команд, формирующих среду для работы:

```
packages("reticulate")
packages("keras")
packages("keras3")
keras::install.keras()
library(keras)
library(keras3)
library(reticulate)
```

Последовательно выполним команды, нажимая сочетание клавиш Ctrl + Enter.



В поле Console (консоль) результатом установки будут следующие строки:

```

> library(reticulate)
Предупреждение:
пакет 'reticulate' был собран под R версии 4.4.2
> library(keras)
Registered 53 methods overwritten by 'keras':
  method from
as.data.frame.keras_training_history keras3
plot.keras_training_history          keras3
print.keras_training_history          keras3
r_to_py.R6ClassGenerator              keras3
Предупреждение:
пакет 'keras' был собран под R версии 4.4.2
> library(keras3)
Registered 53 methods overwritten by 'keras3':
  method from

```

Добавляем новые команды к скрипту. Задаем путь к папке, где находится Python: **venv_path** `file.path(getwd(), "r-reticulate")`, **Ctrl + Enter**.

При выполнении команды в R консоль возвратит значение, которое отобразится в поле команд.



Записываем следующую команду, которая активирует виртуальную среду: **use_virtualenv(venv_path, required = TRUE)**, **Ctrl +**

Enter. Параметр «required = TRUE» позволяет выявить ошибку, если среда не будет успешно активирована.

```

2 use_virtualenv(venv_path, required = TRUE)

```

Последней командой проверяем, установлены ли необходимые библиотеки и связи: **py_config()**, **Ctrl + Enter**.

В консоли выводится список с указанием

информации об установленных программах и библиотеках. В случае успешной установки среды Keras все 9 строк должны быть заполнены текстом без указания ошибки.

```

> py_config()
python: C:/Users/marfi/Documents/.virtualenvs/r-tensorflow/Scripts/python.exe
libpython: C:/Python/312/python312.dll
pythonhome: C:/Users/marfi/Documents/.virtualenvs/r-tensorflow
version: 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
Architecture: 64bit
numpy: C:/Users/marfi/Documents/.virtualenvs/r-tensorflow/Lib/site-packages/numpy
numpy_version: 2.0.2
keras: C:/Users/marfi/DOCUHE-1/VIRTUA-1/R-TENS-1/Lib/site-packages/keras/_init_.p
NOTE: Python version was forced by import("keras")

```

Если к этому этапу установка прошла успешно, можно сразу перейти к этапу «Использование Keras».

Однако при построении виртуальной среды в RStudio могут возникнуть ошибки, связанные с отсутствием «установленных» библиотек. В этом случае рекомендуется создать новую виртуальную среду с нуля и повторить процесс установки библиотек. Если команда **py_config()** не предоставляет полную информацию об установленных библиотек, можно использовать следующий алгоритм. В новом скрипте прописываем команду автоматической установки самых необходимых библиотек и выполняем ее (**Ctrl + Enter**):

py_install(c("numpy", "keras"), env = venv_path)

py_config()

Если после выполнения **py_config()** в R отображается, что библиотеки не найдены, это может быть связано с использованием R другой интерпретации Python или несоответствием виртуальных путей. В этом случае необходимо явно указать свой путь, где находятся установленные библиотеки, добавив в новый скрипт команду:

use_python("C:/Windows/system32/r-reticulate/Scripts/python.exe", required = TRUE)

py_config()

и выполнить ее (**Ctrl + Enter**).

С помощью дополнительной команды **py_run_string("import numpy; import keras")** следует проверить наличие библиотек.

Отметим, что создание виртуальной среды с указанием пути в форме C:\Windows\system32\ может быть некорректным, т. к. этот путь зарезервирован для системных файлов Windows. Запуск интерпретатора или установка пакетов в этом каталоге могут привести к проблемам с правами доступа или конфликтам. Рекомендуется создать отдельную виртуальную среду или использовать установленный Python в другом каталоге (например, в AppData), чтобы избежать возможных проблем.

Если в остальных случаях программа не обнаруживает библиотеки, то существует еще один алгоритм, который также поможет создать, активировать другую среду и установить библиотеки. Вызываем командную строку (см. выше: Win + R, powershell, OK). Вместо команды **python -m venv r-reticulate** вводим **python -m venv myenv**, Enter.

Активируем окружение командой **myenv\Scripts\activate**, Enter.

Устанавливаем библиотеки **pip install numpy keras tensorflow**, Enter. Начнется процесс инсталляции.

В Rstudio в скрипте последовательно выполним команды сочетанием клавиш Ctrl +

Enter для активации новой виртуальной среды и проверки установленных библиотек.

library(reticulate)

use_virtualenv("D:/myenv", required = TRUE)

py_config()

и выполнить ее (Ctrl + Enter).

С помощью дополнительной команды **py_run_string("import numpy; import keras")** следует проверить наличие библиотек.

Отметим, что создание виртуальной среды с указанием пути в форме C:\Windows\system32\ может быть некорректным, т. к. этот путь зарезервирован для системных файлов Windows. Запуск интерпретатора или установка пакетов в этом каталоге могут привести к проблемам с правами доступа или конфликтам. Рекомендуется создать отдельную виртуальную среду или использовать установленный Python в другом каталоге (например, в AppData), чтобы избежать возможных проблем.

Если в остальных случаях программа не обнаруживает библиотеки, то существует еще один алгоритм, который также поможет создать, активировать другую среду и установить библиотеки. Вызываем командную строку (см. выше: Win + R, powershell, OK). Вместо команды **python -m venv r-reticulate** вводим **python -m venv myenv**, Enter.



Активируем окружение командой **myenv\Scripts\activate**, Enter.

```
PS C:\Users\marfi> myenv\Scripts\activate
```

Устанавливаем библиотеки **pip install numpy keras tensorflow**, Enter. Начнется процесс инсталляции.

```
PS C:\Users\marfi> pip install numpy keras tensorflow
```

В Rstudio в скрипте последовательно выполним команды сочетанием клавиш Ctrl + Enter для активации новой виртуальной среды и проверки установленных библиотек.

```
library(reticulate)
use_virtualenv("D:/myenv", required=TRUE)
py_config()
```

Использование Keras

Установленные программы и библиотеки необходимо загружать при выполнении любого скрипта R с помощью функции **library()**.

Подготовка данных

В среде Keras *массивы* данных принято называть *тензорами*. Смысл такой замены терминов здесь не важен, поскольку в прикладном плане ничего не меняется. По-прежнему задачей моделирования является классификация пробных площадок – строк двумерного массива (объектов тензора).

В расчетах использованы данные для 40 пробных площадок (120 строк) и 10 столбцов (9 каналов спектральной яркости и вектор с

номером кластера) (таблица [bancl.csv](#), массив **pcl**). Расчет и построение нейросетевой модели выполнены в скрипте [modelkeras.R](#).

Обучающий тензор, **train_data**, содержал 70 % данных (28 площадок, 84 строки). Тестовый тензор, **test_data**, включал оставшиеся 30 % (12 площадок, 36 строк). Разные показатели имели отличающиеся диапазоны значений; для них выполнили преобразование в безразмерные величины из интервала [0, 1] с помощью формулы $z = (\max(x) - x) / (\max(x) - \min(x))$. Трансформация каждого признака проведена отдельно для обучающей и тестовой выборки.

В среде Keras индексы массивов начи-

наются с нуля. Для преобразования меток классов в формат, пригодный для обучения моделей, номера кластеров от 1 до 4 были уменьшены на единицу и стали номерами от 0 до 3. Далее с помощью функции категориального кодирования **to_categorical()** создана бинарная матрица **train_labels**, состоящая из 4 столбцов (идентифицирующая индекс кластера) и 84 строк (соответствующих объектам, 28 площадкам). Столбец 1 соответствует классу 0, столбец 2 – классу 1 и т. д. Для каждого объекта (строки) назначается индекс 1 в том столбце, который соответствует номеру его кластера, и 0 – в остальных столбцах (табл. 1).

Таблица 1. Фрагмент-матрица **train_labels** для идентификации точек с площадок по кластерам. Преобразованные номера кластера

Номер точки площадки	Номер кластера	Индекс кластеров			
		0	1	2	3
1	2	0	0	1	0
2	1	0	1	0	0
3	2	0	0	1	0
4	2	0	0	1	0
5	2	0	0	1	0
6	3	0	0	0	1
7	3	0	0	0	1
8	0	1	0	0	0
9	1	0	1	0	0
10	1	0	1	0	0
11	0	1	0	0	0
12	1	0	1	0	0
...

Процедура моделирования

Работа с сетью проходила в несколько этапов: подготовка данных, создание модели, подготовка к обучению (компиляция), обучение (Коросов, 2023).

Вначале задали архитектуру нейронной сети (полносвязный персептрон) с помощью функции **keras_model_sequential()** %>% . В разных вариантах модели задавали от 1 до 3 слоев и от 10 до 60 нейронов, в примере показана двухслойная модель с 20 и 10 нейронами. Функция активации нейронов принята **relu**, которая широко применяется благодаря своей простоте (Балута и др., 2023). Поскольку решается многоклассовая задача, на последнем (выходном) слое в модели за-

дается функция активации **softmax**. На выходе получаем 4-мерный вектор вероятности принадлежности каждого образца к каждому из 4 кластеров.

```
model <- keras_model_sequential() %>%
  layer_dense(units = 20, activation = 'relu') %>%
  layer_dense(units = 10, activation = 'relu') %>%
  layer_dense(units = 4, activation = 'softmax')
```

Визуализация структуры модели и потоков информации через слои нейронной сети в виде схемы выполняется с помощью команды **plot(model, show_shapes = TRUE)** (рис. 1). В примере 9 входных нейронов (None, 9) передают данные в 20 нейронов второго (скрытого) слоя (None, 20), а те – в 10 нейронов третьего слоя (None, 10).

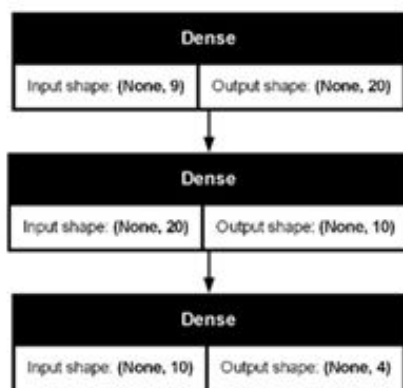


Рис. 1. Вариант графа модели с 3 полносвязными слоями Dense с информацией о размере
Fig. 1. A variant of the model graph with 3 fully connected Dense layers with size information

Далее загружается код в память компьютера (компиляция) для последующей настройки. Для этого в функции **compile()** задаются следующие аргументы: функция потерь **categorical_crossentropy**, алгоритм оптимизации **adam** и метрика точности (успешности) **accuracy** классификации. Функция потерь **categorical_crossentropy** применяется именно для решения многоклассовых задач.

```
model %>% compile(
  optimizer = 'adam',
  loss = 'categorical_crossentropy',
  metrics = c('accuracy'))
```

Обучение сети (training) проводилось на входных данных тензора **train_data** (яркостные характеристики 9 каналов), в качестве целевых переменных служили метки кластеров **train_labels** (4 кластера). В процессе настройки подбиралось число эпох (лучший вариант – 40) и число итераций (10). Применялся метод **fit()** с аргументом **validation_data = 0.2** (20 %). Для оценки степени ее обобщенности, т. е. способности модели работать с новыми данными, принято определенную часть обучающих данных использовать только для вычисления метрики потерь (Шолле, 2022). Оценку (validation) обученной модели проводили, используя метод **evaluate()**. Сначала модель была оценена на тестовых данных **test_data** с соответствующими метками классов **test_labels**. Далее были получены метрики точности (**accuracy**) и функции потерь (**loss**) для всего тестового набора.

```
history<-model%>% fit(train_data, train_labels, epochs = 40, batch_size = 10, validation_split = 0.2)
plot(model, show_shapes = TRUE)
plot(history)
model %>% evaluate(test_data, test_labels)
```

В ходе обучения параметры модели нейронной сети стабилизируются уже через 30 эпох (рис. 2., **plot(history)**). Точность моделей на обучающих и тестовых выборках варьировалась в диапазонах от 55 до 90 %. Увеличение размеров сети (с 5 до 30 нейронов) существенно улучшило прогнозные свойства модели; точность превышает 80 %.

Полученная модель использовалась для расчета типа кластера (местообитания) для каждого пикселя геоизображения (спутниковые снимки) всей изучаемой территории.

Построение прогноза и экстраполяция

Как было описано выше, полный преобразованный набор яркостных характеристик геоизображения содержится в массиве **fulldata**. Используя сигнатуру (соотношение 9 значений яркости) отдельного пикселя, модель рассчитывает для него четыре значения вероятностей принадлежности к одному из 4 классов: **predinew<-model%>% predict(as.matrix(fulldata))**. Эти вероятности в сумме дают 1. Далее из них с помощью функции **preclas** выбирается максимальное значение и данному пикселю приписывается номер столбца как класс наиболее вероятного типа биотопа. Каждый образец получает единственный предсказанный класс (от 1 до 4) (табл. 2). Например, для пикселя 10 получили вектор [0.112, 0.702, 0.151, 0.048] и назначаем класс 1, это разреженные вырубки.

В качестве визуализации предсказаний используется растр с исходной геопривязкой (например, первый канал, **ma**), в который переносятся предсказанные значения типов местообитания **values(ma)**. Далее построили растровый слой по категориям (функция **classify** обеспечивает раскраску значений).

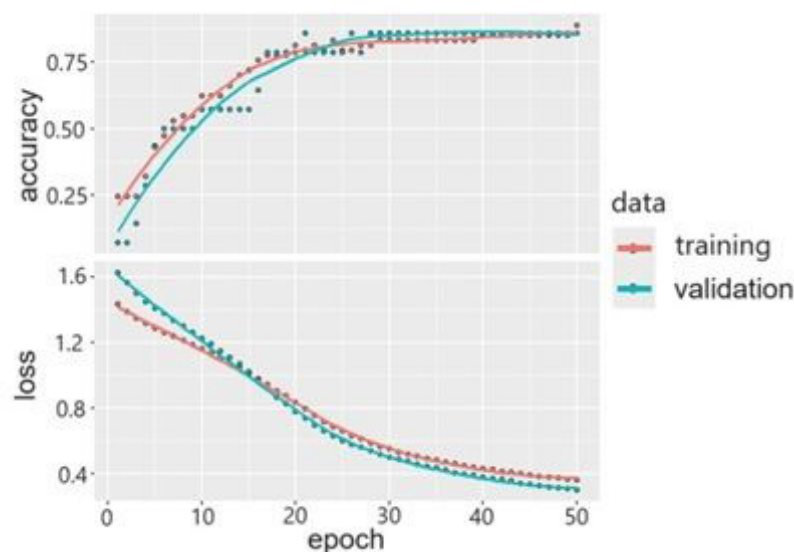


Рис. 2. Динамика оценок потерь (loss) и точности (accuracy) нейросетевой модели при обучении
Fig. 2. Dynamics of loss and accuracy estimates of the neural network model during training

Таблица 2. Фрагмент матрицы прогнозов для построения карты

Номер пикселя	Номер кластера			
	0	1	2	3
1	0.247	0.188	0.384	0.18
2	0.204	0.244	0.419	0.132
...
10	0.112	0.702	0.151	0.048
...

В результате создается карта, отображающая предсказанные классы биотопов (рис. 3).

Однородные области, сформированные пикселями одного кластера, интерпретируются как отдельные типы биотопов. На итоговой псевдоцветной карте показано 5 кластеров: четыре соответствуют изученным в полевых условиях типам, пятый объединяет неисследованные объекты, которые соответствуют спелым и припевающим древо-стоям. Крупные элементы антропогенного и природного происхождения (озера, болота, луга, населенные пункты) закрыли маскирующими белыми полигонами.

Типичная процедура верификации результирующих карт в основном выполняется при сопоставлении их с разными источниками данных. В первую очередь с полевыми геоботаническими описаниями, сопровождаемыми GPS-привязкой и фотографированием местности; с космическими снимками высокого разрешения (Лавриненко, 2015; Раевский и др., 2022). Помимо полевых данных, для верификации используют

лесоустроительные материалы. Также создание векторных карт на основе дешифрирования ДЗ с применением алгоритмов контролируемой классификации, например таких, как метод минимального расстояния, метод Махаланобиса, метод максимального правдоподобия, к спектральному снимкам среднего разрешения с последующей углубленной постклассификационной обработкой, с применением обучаемых алгоритмов Random Forest (Раевский и др., 2022; Семакина и др., 2025). Использование аэрофотоснимков и данных БПЛА тоже служит одним из верификационных показателей. Особенно перспективным является использование сверточных нейронных сетей для автоматического распознавания на снимках растительных сообществ, которые на порядки повышают точность создаваемых карт, что, в свою очередь, кардинально улучшает качество оценки актуального состояния и динамики растительного покрова (Лавриненко, 2023; Элешкевич и др., 2023). Поскольку наша статья носит методический

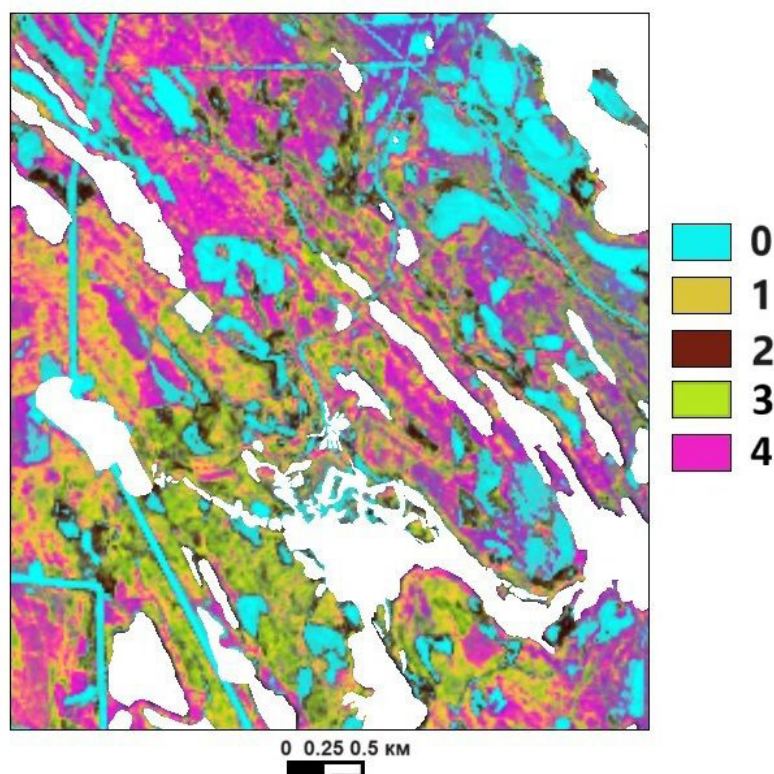


Рис. 3. Грид типов биотопов, построенный на основе двухслойного персептрона ($20 > 10 > 5$); 0 – открытые вырубki, 1 – разреженные вырубki, 2 – молодняки, 3 – лиственные леса, 4 – хвойные и смешанные леса; белыми полигонами закрыты болота, луга, озера, населенные пункты

Fig. 3. Grid of biotope types constructed on the basis of a two-layer perceptron ($20 > 10 > 5$); 0 – open clearings, 1 – sparse clearings, 2 – young forests, 3 – deciduous forests, 4 – coniferous and mixed forests); white polygons indicate swamps, meadows, lakes, and populated areas

характер и относится в основном к практике применения пакета Keras, мы не стали применять сложные процедуры верификации, требующие развернутой интерпретации.

Проверку соответствия грид-карты реальных объектов выполнили визуально, сопоставляя с [космическими снимками высокого разрешения](#), фотографиями местности и полевыми описаниями. Используя последний вариант сети, контуры ландшафтов удалось выделить достаточно хорошо. Четко идентифицировались свежие и разреженные вырубki (0, 1), лиственный лес (3) с молодняками (2), спелые хвойные и смешанные леса (4). Из-за отсутствия объемного массива данных провели приблизительную верификацию на основе трех точек, отобранных из новых геоботанических описаний, выполненных в июне 2025 г. Сравнение карты с этими описаниями показало, что она довольно хорошо отражает реальность. Первая точка соответствовала хвойным лесам с преобладанием сосны (плотность – 24.5), вторая – свежим вырубкам с нулевой плотностью, третья – лиственным лесам с преобладанием березы и ели (14.5 и 8.5).

Обсуждение

Использование библиотеки Keras оказалось более эффективным, чем пакета neuralnet. В наших вычислительных экспериментах, включавших сотни прогонов с различными схемами сетей (от 3 до 50 нейронов в слоях, от 1 до 3 скрытых слоев) и комбинациями летних и зимних каналов (от 3 до 11), Keras обеспечил сокращение ошибки классификации на 20 % при существенном сокращении времени обучения (табл. 3).

Оптимизация архитектуры сети и предобработка данных выявили специфические методологические проблемы, характерные для экологических исследований с ограниченным числом репрезентативных площадок. Увеличение сложности сети до формата 20-10-4 снижало ошибку обучения. Лучший результат с помощью Keras достигнут при наборе данных № 4 (см. табл. 3) – обучающая точность составила 85 %, тестовая – 83 %; продолжительность обучения – всего около 0.09 минуты (~5.4 секунды). Однако дальнейшее усложнение сети привело к переобучению, что выразилось в расхождении точности прогноза на тренировочной (> 90 %) и

Таблица 3. Характеристики процесса обучения нейронных сетей для двух библиотек

№	Число слоев	Число нейронов	Среднее время обучения, мин	Средняя доля совпадений для обучающей выборки	Средняя доля совпадений для тестовой выборки
neuralnet					
1	1	5	0.53	0.39	0.43
2	1	20	1.45	0.61	0.75
3	1	30	3.83	0.65	0.67
4	2	20, 10	6.67	0.99	0.88
5	3	34, 12, 4	7	0.99	0.75
Keras					
1	1	5	0.08	0.54	0.79
2	1	20	0.07	0.82	0.72
3	1	30	0.08	0.79	0.72
4	2	20,10	0.09	0.85	0.83
5	3	34, 12, 4	0.08	0.91	0.79

тестовой (< 70 %) выборках. Использовалось различное число (от 3 до 11) и соотношение числа каналов (от 1 до 5) летних и зимних снимков (табл. 4). Увеличение количества зимних каналов, чувствительных к наличию хвои и структуре снежного покрова, не всег-

да приводило к ожидаемому улучшению классификации вторичных листовенных биотопов; в некоторых конфигурациях происходило «размывание» спектральных сигнатур для кластеров (биотопической картины) (Со-чилова, Ершов, 2012).

Таблица 4. Влияние архитектурных гиперпараметров на точность модели

Количество летних и зимних каналов	Слои	Нейроны	Средняя точность на тестовой и обучающей выборках
7	1	30	80 и 72
7 и 1	2	30 и 20	85 и 81
7	2	20 и 10	78 и 70
7 и 2	2	20 и 10	87 и 87

На наш взгляд, это обусловлено доминированием листовенных пород на изучаемых вырубках и относительно высокой выраженностью сигнала хвойных в зимних каналах Landsat 8 на данной стадии сукцессии, что подчеркивает необходимость подхода к выбору важных сезонных каналов под конкретные экологические задачи. Комбинация 7 летних и 2 зимних каналов оказалась оптимальной, т. к. летние каналы несли основную информацию о состоянии растительного покрова, критически важную для различения стадий сукцессии, а ограниченное число зимних каналов добавляло умеренный контраст, не перегружая модель избыточными или конфликтующими сигналами.

Существенной стороной наших исследований явились выявления естественной типологии изучаемых местообитаний на

основе совместной классификации полевых и спутниковых данных. Как справедливо отмечают авторы: «Метод управляемой классификации является... субъективным» (Данилова и др., 2017, с. 12), поскольку назначение типа биотопа (насаждений) выполняется самим исследователем, исходя из интуитивных представлений о типологии и наличных (часто фрагментарных, разноречивых, устаревших...) данных. Закрепление за определенным выделом того или иного статуса всегда будет контекстуальным из-за ограниченного объема доступной информации, а также связано с большими временными потерями.

Вследствие этого мы задались вопросом, если при дешифрировании объективность априори недостижима, нельзя ли упростить, ускорить и автоматизировать процедуру

выявления природной типологии на основе любой имеющейся информации. Таким приемом стала совместная обработка материалов полевого описания и спутниковых данных. Существенным моментом выполненной кластеризации и выявления типов биотопов стала возможность статистического обобщения характеристик для каждого из выявленных типов. Тем самым мы получили основание для содержательной интерпретации выявленных типов местообитаний, для определения их «естественного» типологического статуса. Рассчитанная на этой основе экстраполяция типов биотопов на всю изучаемую территорию в определенной степени сохраняет природную континуальность свойств среды, не искаженную априорными соображениями исследователя. Эта особенность анализа проявилась, в частности, в том, что на территории разновозрастных вырубок обнаружилась внутренняя неоднородность насаждений. Улавливание таких скрытых закономерностей принципиально важно в экологии животных, где точность описания гетерогенности местообитаний напрямую влияет на понимание закономерностей пространственного распределения и динамики популяций.

Мы рассчитываем, что наша публикация перенесет применение библиотеки Keras в экологических исследованиях из узкой области прикладных задач (таких как мониторинг антропогенных выбросов – Косулин, 2023) на более широкое поле поиска биолого-экологических градиентов.

Заключение или выводы

Библиотека алгоритмов глубокого обучения Keras практически не используется в работах по экологии животных, хотя она позволяет существенно усилить вычислительные

способности пакета R при обработке пространственной и спутниковой информации с целью изучения местообитаний животных. Наш опыт выявления типологии разнородных биотопов и создания грид-карты привел к следующим выводам.

1. Одновременное использование яркостных характеристик спутниковых снимков и полевых геоботанических описаний в процессе кластеризации пробных площадок позволяет в автоматическом режиме построить «естественную» типологию местообитаний (биотопов).
2. Степень дифференциации и точность полученной грид-карты определяются полнотой доступных эколого-биотопических и дистанционных характеристик местообитаний.
3. Результат классификации космического снимка, точность выявления типа растительности зависят от сочетания использованной комбинации каналов зимних и летних снимков, которые в разной степени отражают физиономические свойства растительности. В наших условиях минимально необходимая комбинация из семи летних и двух зимних каналов дала лучший результат.
4. Эффективность классификации определяется структурой нейронной сети, количеством нейронов в скрытых слоях, которые приходится подбирать вручную. Небольшое число нейронов, которым оперирует функция *neuralnet*, не позволяет за разумное время получить точную настройку. Использование слишком больших сетей в моделях Keras приводит к переобучению.
5. Явным преимуществом пакета Keras является возможность обработки больших массивов данных за короткое время.

Библиография

- Балута В. И., Осипов В. П., Рыков Ю. Г., Четверушкин Б. Н. О понятии влияния в концепции когнитивного моделирования при использовании функции активации типа ReLU // Информационные технологии и вычислительные системы. 2023. № 4. С. 59–71. URL: https://www.elibrary.ru/download/elibrary_56573800_56563851.pdf (дата обращения: 08.09.2025).
- Бугмырин С. Н., Коросов А. В., Иешко Е. П., Аниканова В. С., Беспятова Л. А., Матросова Ю. М., Телегин И. В. Опыт изучения пространственного распределения паразитов мелких млекопитающих // Северная Европа в XXI веке: Природа, культура, экономика: Материалы междунар. конф., посвящ. 60-летию КарНЦ РАН (24–27 октября 2006). Петрозаводск, 2006. С. 55–58.
- Геоинформатика: Учеб. для студентов вузов / Е. Г. Капралов, А. В. Кошкарёв, В. С. Тикунов и др.; Под ред. В. С. Тикунова. М.: Изд. центр «Академия», 2005. 480 с.
- Гусева Т. Л., Коросов А. В., Беспятова Л. А., Аниканова В. С. Многолетняя динамика биотопического размещения обыкновенной бурозубки (*Sorex araneus*, Linnaeus 1758) в мозаичных ландшафтах Карелии // Ученые записки Петрозаводского государственного университета. 2014. Т. 2, № 8. С. 13–20. URL: <https://cyberleninka.ru/article/n/mnogoletnyaya-dinamika->

- [biotopicheskogo-razmescheniya-obyknovennoy-burozubki-sorex-araneus-linnaeus-1758-v-mozaichnyh-landshaftah/viewer](#) (дата обращения: 10.03.2025).
- Данилова И. В., Корец М. А., Рыжкова В. А. Картографирование возрастных стадий лесной растительности на основе анализа разносезонных спутниковых изображений Landsat // Исследование земли из космоса. 2017. № 4. С. 12–24. URL: https://elibrary.ru/download/elibrary_29992173_79574715.pdf (дата обращения: 10.03.2025).
- Иешко Е. П., Коросов А. В., Никонорова И. А., Бугмырин С. В. Связь видового богатства сообществ гельминтов и численности хозяина (на примере обыкновенной бурозубки *Soxrex araneus*) // Паразитология. 2020. Т. 54, № 1. С. 3–12.
- Ильчук М. А., Цай С. С. Формирование базы данных эталонно-калибровочных участков для тематического дешифрирования материалов космической съемки // Труды БГТУ. Серия 1. Лесное хозяйство. 2010. Вып. 18. С. 44–47.
- Раевский Б. В., Тарасенко В. В., Петров Н. В. Оценка современного состояния растительных сообществ заповедника «Костомукшский» по спутниковым снимкам системы Landsat // Современные проблемы дистанционного зондирования Земли из космоса. 2022. Т. 19, № 3. С. 47–61. URL: <http://jr.rse.cosmos.ru/article.aspx?id=2528> (дата обращения: 10.11.2025).
- Коросов А. В., Матросова Ю. М., Бугмырин С. В., Аниканова В. С., Беспятова Л. А. Опыт реконструкции территориального размещения рыжей полевки в мозаичном ландшафте южной Карелии // Биogeография Карелии (флора и фауна таежных экосистем). Вып. 4. Петрозаводск, 2003. С. 204–212.
- Коросов А. В. Нейронные сети для экологии: введение // Принципы экологии. 2023. № 3. С. 76–96. DOI: 10.15393/j1.art.2023.14002 (дата обращения: 26.02.2025).
- Коросов А. В., Марфицына Н. А. Дешифрирование местообитаний животных с помощью методов глубокого обучения библиотеки Keras // ИнтерКарта. ИнтерГИС: Материалы междунар. конф. Пермь, 2025. (В печати)
- Косулин В. В. Применение систем искусственного интеллекта в экологическом мониторинге выбросов предприятия // Научно-технический вестник Поволжья. 2023. № 10. С. 132–135. URL: [elibrary_54795395_27315499.pdf](#) (дата обращения: 26.02.2025).
- Кузьменко Е. И., Фролов А. А., Силаев А. В. Картирование лесных ландшафтов северо-запада Западной Сибири с использованием ГИС // География и природные ресурсы. 2015. № 4. С. 151–161.
- Лавриненко И. А. Типология территориальных единиц растительности для целей крупномасштабного картографирования (на примере острова Колгуев) // Геоботаническое картографирование. 2015. С. 95–119. URL: <https://cyberleninka.ru/article/n/tipologiya-territorialnyh-edinit-rastitelnosti-dlya-tseley-kрупномasshtabnogo-kartografirovaniya-na-primere-ostrova-kolguev> (дата обращения: 12.12.2025).
- Лавриненко И. А. Современные подходы к подготовке карт растительности Арктики // Геоботаническое картографирование. 2023. С. 14–27. URL: <https://cyberleninka.ru/article/n/sovremennye-podhody-k-podgotovke-kart-rastitelnosti-arktiki> (дата обращения: 12.12.2025).
- Семакина А. В., Шихов А. Н., Климина Е. А. Картографирование уязвимости лесов к ветровому воздействию на основе спутниковых данных (на примере Пермского края) // Современные проблемы дистанционного зондирования Земли из космоса. 2025. Т. 22, № 4. С. 236–252.
- Сочилова Е. Н., Ершов Д. В. Анализ возможности определения запасов древесных пород по спутниковым данным Landsat ETM+ // Современные проблемы дистанционного зондирования Земли из космоса. 2012. Т. 9, № 3. С. 277–282. URL: <http://jr.rse.cosmos.ru/article.aspx?id=1076> (дата обращения: 26.02.2025).
- Шитиков В. К., Мاستицкий С. Э. Классификация, регрессия и другие алгоритмы Data Mining с использованием R. 2017. 351 с. URL: <https://github.com/ranalytics/data-mining> (дата обращения: 26.01.2025).
- Шолле Ф. Глубокое обучение с R и Keras. М.: ДМК Пресс, 2022. 646 с. URL: <https://coollib.net/b/627871-fransua-sholle-glubokoe-obuchenie-s-r-i-keras> (дата обращения: 26.02.2025).
- Элешкевич А. Д., Еременко М. С., Сайбель Е. Г., Христолюбов И. А., Чернов А. Г. Применение методов автоматизированного дешифрирования в задачах распознавания зон антропогенного воздействия на нефтегазовых месторождениях // Экспозиция Нефть Газ. 2023. № 7. С. 127–131. DOI: 10.24412/2076-6785-2023-7-127-131
- GeeksforGeeks. How to Install Keras in Windows. 2021. URL: <https://www.geeksforgeeks.org/how-to-install-keras-in-windows/?ysclid=m8oygmq0nq900158163> (дата обращения: 15.01.2025).
- PythonRu. 2021. URL: <https://pythonru.com/biblioteki/kak-ustanovit-keras-na-linux-i-windows-keras?ysclid=m8co0qtcos915911868> (дата обращения: 15.01.2025).
- The R Project for Statistical Computing. 2023. URL: <https://www.r-project.org/> (дата обращения: 26.01.2025).

USING THE KERAS LIBRARY TO DECRYPT ANIMAL HABITATS USING DEEP LEARNING METHODS

MARFITSYNA *Petrozavodsk State University, 33 Lenin Ave., Petrozavodsk, Republic of Karelia, 185910, Russia, marfitsyna.nata@mail.ru*
Natalya Alexandrovna

KOROSOV *DSc, Petrozavodsk State University, 33 Lenin Ave., Petrozavodsk, Republic of Karelia, 185910, Russia, korosov@psu.karelia.ru*
Andrey Victorovich

Key words:

habitat
RSD
GIS
neural network
R
Keras

Summary: The paper considers the use of deep learning algorithms from the Keras library to solve the problem of classifying forest clearings of different ages using remote sensing in the R environment. A rather complicated procedure for installing Keras libraries on a computer is considered in detail. The stages of neural simulation and their variations using the R neuralnet package and the Keras environment are described. Satellite images were decoded in the vicinity of Gomselga village (Karelia) using field survey data. The typical decryption algorithm (classification with learning) was supplemented by a joint multidimensional analysis of the brightness characteristics of the image and field geobotanical descriptions. As a result, 4 sets of reference signatures were formed, corresponding to a particular state of regenerating clearings. The neural network (multilayer perceptron) was configured to recognize these types of plantings, and then performed the classification of the remaining pixels of the image for the entire studied area. Based on the analysis of geobotanical descriptions and satellite data, a grid map was created highlighting four main types of habitats: fresh cuttings, regenerating cuttings, young trees, and deciduous forest. Data processing using Keras algorithms significantly speeds up analysis, and makes it possible to increase the number of layers and neurons and detail the grid. In particular, unlike the algorithms of reference decoding, the proposed approach made it possible to identify the heterogeneity of vegetation within the same-age clearings. The results of the work are used to identify heterogeneous animal habitats and the influence of environmental factors on their spatial distribution and abundance.

Reviewer: I. R. Idrisov
A. N. Shihov

Received on: 16 November 2025

Published on: 27 December 2025

References

- Baluta V. I. Osipov V. P. Rykov Yu. G. Chetverushkin B. N. On the notion of influence in the concept of cognitive modeling when using a ReLU-type activation function, *Informacionnye tehnologii i vychislitel'nye sistemy*. 2023. No. 4. P. 59–71. URL: https://www.elibrary.ru/download/elibrary_56573800_56563851.pdf (data obrascheniya: 08.09.2025).
- Bugmyrin S. N. Korosov A. V. Ieshko E. P. Anikanova V. S. Bespyatova L. A. Matrosova Yu. M. Telegin I. V. Experience in studying the spatial distribution of small mammalian parasites, *Severnaya Evropa v XXI veke: Priroda, kul'tura, ekonomika: Materialy mezhdunar. konf., posvyasch. 60-letiyu KarNC RAN (24–27 oktyabrya 2006)*. Petrozavodsk, 2006. P. 55–58.
- Danilova I. V. Korec M. A. Ryzhkova V. A. Mapping of age stages of forest vegetation based on the analysis of multi-seasonal satellite images by Landsat, *Issledovanie zemli iz kosmosa*. 2017. No. 4. P. 12–24. URL: https://elibrary.ru/download/elibrary_29992173_79574715.pdf (data obrascheniya: 10.03.2025).
- Eleshkevich A. D. Eremenko M. S. Saybel' E. G. Hristolyubov I. A. Chernov A. G. Application of automated decryption methods in the tasks of recognizing anthropogenic impact zones in oil and gas fields, *Ekspozitsiya Neft' Gaz*. 2023. No. 7. P. 127–131. DOI: 10.24412/2076-6785-2023-7-127-131
- GeeksforGeeks. How to Install Keras in Windows. 2021. URL: <https://www.geeksforgeeks.org/how-to-install-keras-in-windows/?ysclid=m8oygmq0nq900158163> (data obrascheniya: 15.01.2025).
- Geoinformatics: A textbook for university students, E. G. Kapralov, A. V. Koshkarev, V. P. Tikunov i dr.; Pod red. V. P. Tikunova. M.: Izd. centr «Akademiya», 2005. 480 p.

- Guseva T. L. Korosov A. V. Bespyatova L. A. Anikanova V. S. Long-term dynamics of biotopic distribution of the common shrew (*Sorex araneus*, Linnaeus 1758) in mosaic landscapes of Karelia, *Uchenye zapiski Petrozavodskogo gosudarstvennogo universiteta*. 2014. T. 2, No. 8. P. 13–20. URL: <https://cyberleninka.ru/article/n/mnogoletnyaya-dinamika-biotopicheskogo-razmescheniya-obyknovennoy-burozubki-sorex-araneus-linnaeus-1758-v-mozaichnyh-landshaftah/viewer> (data obrascheniya: 10.03.2025).
- Ieshko E. P. Korosov A. V. Nikonorova I. A. Bugmyrin S. V. The relationship between the species richness of helminth communities and the abundance of the host (using the example of the common shrew *Sorex araneus*), *Parazitologiya*. 2020. T. 54, No. 1. P. 3–12.
- Il'yuchik M. A. Cay S. S. Formation of a database of reference and calibration sites for thematic decoding of space survey materials, *Trudy BGTU. Seriya 1. Lesnoe hozyaystvo*. 2010. Vyp. 18. P. 44–47.
- Korosov A. V. Marficya N. A. Decryption of animal habitats using deep learning methods of Keras Library, *InterKarta. InterGIS: Materialy mezhdunar. konf. Perm'*, 2025. (V pechaty)
- Korosov A. V. Matrosova Yu. M. Bugmyrin S. V. Anikanova B. C. Bespyatova L. A. The experience of reconstruction of the territorial location of the red vole in the mosaic landscape of South Karelia, *Biogeografiya Karelii (flora i fauna taezhnyh ekosistem)*. Vyp. 4. Petrozavodsk, 2003. P. 204–212.
- Korosov A. V. Neural networks for ecology: an introduction, *Principy ekologii*. 2023. No. 3. P. 76–96. DOI: 10.15393/j1.art.2023.14002 (data obrascheniya: 26.02.2025).
- Kosulin V. V. Application of artificial intelligence systems in environmental monitoring of enterprise emissions, *Nauchno-tehnicheskij vestnik Povolzh'ya*. 2023. No. 10. P. 132–135. URL: [elibrary_54795395_27315499.pdf](https://elibrary.ru/54795395_27315499.pdf) (data obrascheniya: 26.02.2025).
- Kuz'menko E. I. Frolov A. A. Silaev A. V. Mapping of forest landscapes in the north-west of Western Siberia using GIS, *Geografiya i prirodnye resursy*. 2015. No. 4. P. 151–161.
- Lavrinenko I. A. Modern approaches to the preparation of maps of Arctic vegetation, *Geobotanicheskoe kartografirovanie*. 2023. P. 14–27. URL: <https://cyberleninka.ru/article/n/sovremennye-podhody-k-podgotovke-kart-rastitelnosti-arktiki> (data obrascheniya: 12.12.2025).
- Lavrinenko I. A. Typology of territorial units of vegetation for the purposes of large-scale mapping (using the example of the Island of Kolguyev), *Geobotanicheskoe kartografirovanie*. 2015. P. 95–119. URL: <https://cyberleninka.ru/article/n/tipologiya-territorialnyh-edinit-rastitelnosti-dlya-tseley-krupnomasshtabnogo-kartografirovaniya-na-primere-ostrova-kolguev> (data obrascheniya: 12.12.2025).
- PythonRu. 2021. URL: <https://pythonru.com/biblioteki/kak-ustanovit-keras-na-linux-i-windows-keras?yclid=m8co0qtcos915911868> (data obrascheniya: 15.01.2025).
- Raevskiy B. V. Tarasenko V. V. Petrov N. V. Assessment of the current state of the plant communities of the Kostomukshsky Nature Reserve using satellite images from the Landsat system, *Sovremennye problemy distancionnogo zondirovaniya Zemli iz kosmosa*. 2022. T. 19, No. 3. P. 47–61. URL: <http://jr.rse.cosmos.ru/article.aspx?id=2528> (data obrascheniya: 10.11.2025).
- Semakina A. V. Shihov A. N. Klimina E. A. Mapping the vulnerability of forests to wind impacts based on satellite data (using the example of the Perm Region), *Sovremennye problemy distancionnogo zondirovaniya Zemli iz kosmosa*. 2025. T. 22, No. 4. P. 236–252.
- Shitikov V. K. Mastickiy S. E. Classification, regression, and other Data Mining algorithms using R. 2017. 351 p. URL: <https://github.com/ranalytics/data-mining> (data obrascheniya: 26.01.2025).
- Sholle F. Deep learning with R and Keras. M.: DMK Press, 2022. 646 p. URL: <https://coollib.net/b/627871-fransua-sholle-glubokoe-obuchenie-s-r-i-keras> (data obrascheniya: 26.02.2025).
- Sochilova E. N. Ershov D. V. Analysis of the possibility of determining tree stocks based on Landsat ETM satellite data+, *Sovremennye problemy distancionnogo zondirovaniya Zemli iz kosmosa*. 2012. T. 9, No. 3. P. 277–282. URL: <http://jr.rse.cosmos.ru/article.aspx?id=1076> (data obrascheniya: 26.02.2025).
- The R Project for Statistical Computing. 2023. URL: <https://www.r-project.org/> (data obrascheniya: 26.01.2025).