



**Издатель**

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

Научный электронный журнал

**ПРИНЦИПЫ ЭКОЛОГИИ**

<https://ecopri.ru>

**№ 1 (51). Март, 2024**

**Главный редактор**

А. В. Коросов

**Редакционный совет**

В. Н. Большаков  
А. В. Воронин  
Э. В. Ивантер  
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. К. Шитиков  
В. Н. Якимов

**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, ул. Ленина, 33.

E-mail: [ecopri@petsu.ru](mailto:ecopri@petsu.ru)

<https://ecopri.ru>





УДК 004.8; 59.087

# ПРИМЕНЕНИЕ ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ПРИ ОБРАБОТКЕ ИЗОБРАЖЕНИЙ С ФОТОЛОВУШЕК: ПРИНЦИПЫ, ПРОГРАММЫ, ПОДХОДЫ

**ОГУРЦОВ**  
Сергей Сергеевич

кандидат биологических наук, Центрально-Лесной государственный заповедник, Институт проблем экологии и эволюции им. А. Н. Северцова РАН (Россия, 172521, Тверская область, Нелидовский городской округ, пос. Заповедный), [etundra@mail.ru](mailto:etundra@mail.ru)

**ЕФРЕМОВ**  
Владислав  
Александрович

Московский физико-технический институт (Национальный исследовательский университет) (Россия, 127576, Московская область, г. Москва), [efremov.va@phystech.edu](mailto:efremov.va@phystech.edu)

**ЛЕУС**  
Андрей Владимирович

кандидат технических наук, Московский физико-технический институт (Национальный исследовательский университет) (Россия, 1410006, Московская область, г. Мытищи), [leus.av@mipt.ru](mailto:leus.av@mipt.ru)

## Ключевые слова:

анализ изображений, детектирование, классификация, компьютерное зрение, машинное обучение, нейронные сети, распознавание образов, фотоловушки

**Аннотация.** Искусственный интеллект (ИИ) все глубже проникает в экологическую науку. Наиболее стремительно это проявляется в исследованиях на основе изображений, например, с дронов или фотоловушек. В настоящем обзоре рассматривается современное развитие исследований с фотоловушками в области применения технологий ИИ, а именно компьютерного зрения и глубокого обучения. Вкратце рассмотрены основные понятия машинного обучения (Machine Learning, ML) и глубоких нейронных сетей, необходимые современному биологу и экологу для понимания обработки и анализа изображений. Разобраны возможности применения ИИ для распознавания образов и поиска объектов на изображениях. Приводится обзор современного программного обеспечения с применением технологий компьютерного зрения и машинного обучения для распознавания фотографий и видеоизображений с фотоловушек, а также краткий обзор открытых наборов данных для обучения ML-моделей. В общем виде рассмотрены восемь ML-программ: MegaDetector, EcoAssist, MLWIC2, Conservation AI, FasterRCNN+InceptionResNetV2, DeepFaune, ClassifyMe, а также первая отечественная разработка от Московского физико-технического института. На основе исследований с фотоловушками в Центрально-Лесном заповеднике проведено тестирование программ, выявлены их преимущества и недостатки. В заключение обсужден потенциал применения ИИ в современных исследованиях с фотоловушками. Настоящий обзор будет полезен как биологам и экологам для общего знакомства с нейронными сетями и их применением в области распознавания образов на изображениях, так и ML-специалистам и программистам для понимания применимости ML-моделей в экологической науке и возможностей их обучения на больших массивах данных.

© Петрозаводский государственный университет

## Получена:

28 января 2024 года

**Подписана к печати:**

02 мая 2024 года

## Введение

За последние 10 лет мы наблюдаем стремительное развитие технологий искусственного интеллекта (ИИ), которые все активнее внедряются в природоохранную биологию и экологию (Qin et al., 2016; Xue et al., 2017; Allan et al., 2018; Kwok et al., 2019; Meek et al., 2020; Kellenberger et al., 2021; Tuia et al., 2022; Binta Islam et al., 2023). В большей степени это проявляется в наиболее динамично развивающихся областях с использованием современной техники, таких как фотоловушки и дроны (Schneider et al., 2018, 2019; Glover-Kapfer et al., 2019; Green et al., 2020; Леус, Ефремов, 2021; Михайлов и др., 2021; Corcoran et al., 2021; Бежавский, 2022; Бизиков и др., 2022; Vélez et al., 2023; Xie et al., 2023). В современных условиях экологические исследования, особенно с фотоловушками,

стремительно входят в область науки о больших данных, т.н. «big data science» (Farley et al., 2018; Shepley et al., 2021; Tuia et al., 2022), продуцируя миллионы изображений в рамках одного проекта (Norouzzadeh et al., 2018; Glover-Kapfer et al., 2019; Schneider et al., 2019; Norouzzadeh et al., 2021; Vélez et al., 2023). Традиционные способы аннотирования или тегирования (т.е. присвоения меток или тегов) фото- и видеоизображений оператором вручную требуют все больше и больше временных затрат, что отмечают многие исследователи (Swinnen et al., 2014; Schneider et al., 2018; Beery et al., 2019; Wei et al., 2020; Norouzzadeh et al., 2021; Tuia et al., 2022; Ефремов и др., 2023а). В связи с этим обработка изображений входит в топ-5 проблем, с которыми сталкиваются исследователи с фотоловушками по всему миру (Glover-Kapfer et al., 2019). Кроме того, природоохранный биология часто требует скорейших выводов и оперативных действий, учитывая стремительное сокращение биоразнообразия и уничтожение местообитаний (Ceballos et al., 2020), что делает трату времени на ручную обработку больших массивов данных иногда попросту недопустимой (Kwok et al., 2019).

Сегодня обработка изображений с фотоловушек и других сенсоров является одной из самых популярных и востребованных областей применения машинного обучения в экологической науке (Tuia et al., 2022). Нейронные сети широко используются для отсеивания пустых кадров (Beery et al., 2018; Willi et al., 2019; Tabak et al., 2020; Бизиков и др., 2022), идентификации видов (Carl et al., 2020; Gomez Villa et al., 2017; Norouzzadeh et al., 2018; Tabak et al., 2019; Willi et al., 2019; Whytock et al., 2021; Binta Islam et al., 2023), определения числа особей (Norouzzadeh et al., 2018, 2021; Schneider et al., 2018; Kellenberger et al., 2021; Михайлов и др., 2021) и их индивидуального распознавания (Bogucki et al., 2018; Schofield et al., 2019; Chen et al., 2020; Schneider et al., 2020; Shi et al., 2023).

В то же время большинство современных российских биологов и экологов еще достаточно плохо знакомы с данным направлением IT-отрасли. Между тем во многих случаях использование технологий ИИ способно значительно упростить процесс обработки данных с фотоловушек и существенно облегчить труд исследователей. В связи с этим мы представляем настоящую обзорную статью, целью которой является освещение современных технологических решений для обработки изображений с фотоловушек с применением технологий ИИ (компьютерного зрения и машинного обучения). В рамках поставленной цели выделены следующие задачи: 1) дать краткую теоретическую информацию о машинном обучении и сверточных нейронных сетях, которая будет полезна в практике современного биолога и эколога, работающего с изображениями; 2) провести обзор современного программного обеспечения (ПО) для автоматического распознавания образов на изображениях с фотоловушек; 3) дать рекомендации по использованию рассмотренного ПО на примере собственных исследований.

## **Аналитический обзор**

В настоящем обзоре мы рассматриваем следующие программные продукты, связанные с ИИ: MegaDetector (Beery et al., 2019) и MegaDetector GUI (Gyurov, 2022), EcoAssist (van Lunteren, 2023), MLWIC2 (Tabak et al., 2020), Conservation AI (Chalmers et al., 2019), FasterRCNN+InceptionResNetV2 (Hui, 2018), DeerFaune (Rigoudy et al., 2023), ClassifyMe (Falzon et al., 2020) и ПО Московского физико-технического института (МФТИ) (Леус, Ефремов, 2021). Отдельное внимание мы также уделили открытым наборам данных для обучения собственных моделей машинного обучения. Большинство программ было протестировано на фотографиях и видеоизображениях с фотоловушек в рамках работы Программы фотомониторинга в Центрально-Лесном государственном природном биосферном заповеднике (ЦЛГЗ) CFNR CAMMON (Central Forest Nature Reserve CAMtrap MONitoring). Указанное ПО обладает различными возможностями и требует разных навыков (в т.ч. в области программирования) для успешной работы. Подробное сравнение некоторых свободных программ и особенности их выбора под различные навыки представлены в специальном онлайн-справочнике (Vélez, Fieberg, 2022). Исчерпывающий перечень и краткое описание практически всех известных на сегодняшний день решений в области применения машинного обучения для обработки изображений с фотоловушек может быть найдено [здесь](#). Мы намеренно не рассматриваем в данном обзоре более комплексные программы для полноценной организации и обработки данных с фотоловушек с применением машинного обучения (например, Wildlife Insights, Agouti, Timelapse и т.п.), потому что им посвящен отдельный обзор (Огурцов и др., 2024).

В качестве основного метода в данной работе использовался анализ литературы, презентаций и видеороликов. Поиск литературы (научных статей, монографий, технических отчетов, диссертаций) и презентаций осуществлялся посредством поисковых запросов в базах данных [Scopus](#), [Web of Science](#), [ResearchGate](#) и системе [Google Scholar](#) по ключевым запросам «artificial intelligence», «machine learning», «computer vision», «deep learning», «neural networks» с приставкой «camera traps».

## **Искусственный интеллект: машинное и глубокое обучение**

Искусственный интеллект (Artificial Intelligence, AI) – это достаточно обширный термин, который включает в себя компьютерные системы, имитирующие человеческий интеллект. Одно из направлений ИИ – это машинное обучение (Machine Learning, ML), которое сосредоточено на создании систем, обучающихся и развивающихся на основе получаемых ими данных (Mohri et al.,

2012). Принято выделять три типа машинного обучения: обучение с учителем (supervised learning), обучение без учителя (unsupervised learning), обучение с подкреплением (reinforcement learning). В рамках данной статьи будет рассматриваться только задача обучения с учителем, цель которой заключается в обучении моделей на помеченных данных таким образом, чтобы в дальнейшем делать прогнозы на ранее не встречавшихся примерах (Tuia et al., 2022). Помеченные данные – это такой набор обучающих примеров, где желаемые выходные сигналы для этих примеров уже известны (Mohri et al., 2012). В случае с фотоловушками входными данными являются сами изображения, а выходными эталонами – метки (т.н. «классы» или «теги») видов животных.

Глубокое обучение (Deep Learning, DL) – это подобласть машинного обучения, где центральным объектом исследования являются глубокие нейронные сети (Hagan et al., 1996; LeCun et al., 2015; Goodfellow et al., 2016), которые берут свое начало в работах по изучению головного мозга у млекопитающих (Hu et al., 2015). Каждый искусственный нейрон в такой сети принимает несколько входных сигналов с разными весами, вычисляет взвешенную сумму этих сигналов, пропускает результат через нелинейную функцию активации и передает его на вход другим нейронам (Hagan et al., 1996). Нейроны обычно располагаются в несколько слоев; нейроны каждого слоя получают входные данные от предыдущего слоя, обрабатывают их и передают свой выходной сигнал следующему слою. Разделяют входной слой, куда подается сигнал, выходной слой, откуда снимается результат, и промежуточные скрытые слои. Глубокая нейронная сеть (Deep Neural Network, DNN) – это нейронная сеть, где более 1 скрытого слоя (т.е. всего больше 3 слоев) (Goodfellow et al., 2016). Как правило, свободными параметрами обучаемой модели являются веса (или связи) между нейронами, которые определяют вклад каждого признака во взвешенную сумму (Norouzzadeh et al., 2021).

### **Сверточные нейронные сети для классификации образов**

В задаче классификации образов целью нейросетевых алгоритмов является прогнозирование категориальных меток классов, к которым принадлежат новые образцы данных, на основе прошлых наблюдений. Разделяют бинарную и мультиклассовую классификации. В первом случае задачей алгоритма является предсказание двух возможных меток класса {0, 1}, а во втором таких возможных классов больше, чем 2 {0, 1, 2, ... N}.

Самая классическая нейронная сеть – это многослойная полносвязная или многослойный перцептрон. В ней в полностью связанном слое каждый нейрон получает входные данные от всех нейронов предыдущего слоя. С другой стороны, в сверточных слоях применяется сверточный фильтр, который обрабатывает данные на входе с заданным смещением и передает результат на вход следующего слоя. Числовые значения фильтра – это настраиваемые параметры, которые подбираются во время обучения нейронной сети с целью обнаружения эффективного паттерна (Hagan et al., 1996; Goodfellow et al., 2016). Нейронная сеть с одним или несколькими сверточными слоями называется сверточной нейронной сетью (Convolutional Neural Network, CNN; LeCun et al., 1989), а если таких слоев 3 или больше, то это уже глубокая CNN (deep CNN). Глубокие CNN показали отличные результаты при решении задач, связанных с поиском объектов на изображениях (детектированием) и их распознаванием (классификацией) (LeCun et al., 2015; Goodfellow et al., 2016; Schneider et al., 2018; Vélez et al., 2023), и являются на сегодняшний день самыми популярными в области распознавания образов (Willi et al., 2019).

При помощи сверточных нейронных сетей из входного изображения создаются карты признаков, где каждый элемент соответствует небольшому участку пикселей на исходном изображении. При таком подходе существенно снижается количество обучаемых параметров в отличие от полносвязного перцептрона, где нейрон связан с каждым пикселем исходного изображения. Как правило, размер участка, на который накладывается фильтр / ядро / матрица весовых коэффициентов, имеет размер 3 × 3, 5 × 5, 7 × 7 пикселей. Этот фильтр с теми же значениями весовых коэффициентов смещается вдоль изображения, и таким образом генерируется карта признаков. Чем больше генерируется количество выходных карт, тем больше паттернов (шаблонов) будет захватывать нейронная сеть.

Когда фильтр накладывается на область изображения, значения его коэффициентов умножаются на соответствующие значения пикселей, а затем складываются между собой. Процесс «сканирования» (наложение матрицы фильтра на матрицу изображения с расчетом значений) называется сверткой (LeCun et al., 2015; рис. 1). Это можно сравнить с применением фильтров в графическом редакторе (например, «размытие», «карандаш», «губка» и т.п. в Adobe Photoshop). Для цветных изображений все эти операции выполняются для каждой компоненты (Red, Green, Blue; RGB) с помощью своего фильтра (LeCun et al., 2015).

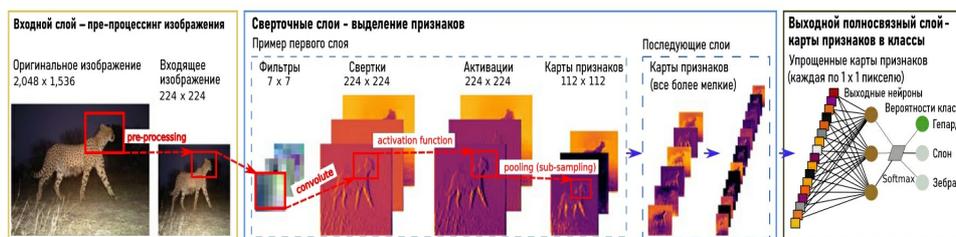


Рис. 1. Схематичное изображение принципа работы сверточной нейронной сети. Взято и переведено из Willi et al., 2019 с дополнениями. Все пояснения приводятся в тексте

Fig. 1. Schematic illustration of a CNN architecture. Taken and translated from Willi et al., 2019 with additions. All explanations are provided in the text

Фильтры позволяют выделять определенные паттерны или закономерности (наклонные линии, контрастные переходы, контуры) на разных участках изображения в соответствии с конфигурацией весовых коэффициентов. На выходе после применения фильтра, а затем нелинейной функции активации формируется карта признаков (канал), которая отражает наличие паттернов на изображении. Совокупность таких карт определяет уникальность (как класса) образа на изображении, по которому в дальнейшем и будет вестись распознавание (см. рис. 1).

Процесс распознавания признаков происходит от общих к частным (Norouzzadeh et al., 2021): ранние слои учатся обнаруживать наиболее генеральные и простые паттерны (например, контуры или края), а последующие слои выявляют уже более сложные и специфические паттерны (Krizhevsky et al., 2012). Поэтому процесс обучения нейросети является восходящим иерархическим (LeCun et al., 2015), т.е. он идет от более простых деталей (линий разного наклона, градиентов) к более сложным (формы, рисунки шкуры, целые фрагменты тела животного). Одно из важнейших свойств свертки является ее инвариантность (т.е. устойчивость) относительно переноса. Это означает, что величина выходного сигнала зависит не от местоположения паттернов на картинке, а от их наличия.

Помимо самой свертки в CNN существует еще одна важная операция – пулинг (pooling) или подвыборка (sub-sampling). Ее суть заключается в автоматическом изменении масштаба: уменьшении размера карты признаков за счет того, что оставляются значения по определенной маске (LeCun et al., 2015). Например, по маске  $2 \times 2$  пикселя с сохранением самого максимального (MaxPooling) или среднего значения (AveragePooling) становится возможным сохранять только наиболее важные признаки. Применяя операцию несколько раз, можно уменьшить изображение в 2, 4, 8 и т.д. раз в зависимости от числа пулингов. Благодаря пулингу удается проводить анализ на более крупном масштабе и выделять на последующих слоях нейронов более общие (важные) признаки. Например, однократный пулинг уменьшает разрешение изображения вдвое (с  $224 \times 224$  до  $112 \times 112$  пикселей; см. рис. 1). Так удастся значительно сократить число обучаемых параметров нейронной сети, что позволяет быстрее проводить ее обучение. Помимо этого избыточное число параметров может приводить к переобучению.

Когда достигается необходимый уровень масштаба карты признаков, она подается на вход заключительного выходного слоя полносвязной нейронной сети (обычного многослойного перцептрона), где и проводится итоговая классификация и рассчитывается вероятность принадлежности объекта на изображении к тому или иному классу (см. рис. 1; LeCun et al., 2015). Для классификации обычно используется логистическая функция (softmax) с выходными значениями от 0 до 1 для каждого класса и с суммой всех выходов, равной 1 (Norouzzadeh et al., 2018). Эти выходы интерпретируются как предполагаемая вероятность принадлежности изображения к определенному классу. Чем она выше, тем больше нейросеть уверена в том, что изображение относится к данному классу (Bridle, 1990).

Связи между слоями в нейронных сетях характеризуются весовыми коэффициентами (параметрами), которые определяют, как нейронная сеть преобразует свои входы в выходы. Обучение нейронной сети означает настройку этих параметров для каждого нейрона таким образом, чтобы вся сеть выдавала желаемый выход для каждого входного примера. Для такой настройки вычисляется мера расхождения между текущим выходом сети и желаемым выходом; эта мера расхождения называется функцией потерь. Функция потерь оценивает разницу между предсказанным выходом (видом животного, который нейросеть считает наиболее вероятным) и истинным выходом (меткой / тегом вида на изображении, т.е. выходом, который мы хотим получить). Затем при помощи алгоритма обратного распространения ошибки (backpropagation) все весовые коэффициенты нейронной сети обновляются таким образом, чтобы минимизировать функцию потерь (LeCun et al., 2015). Используемый алгоритм является итерационным, т.е. в процессе обучения он применяется многократно. На каждой итерации происходит обновление весовых коэффициентов, причем зачастую за одну итерацию проходит не одно, а пакет изображений, что позволяет ускорять как обучение нейронной сети, так и ее сходимость.

Число сверточных слоев и характер связи нейронов внутри них и между ними определяет

архитектуру (т.е. устройство) нейросети. Для задач классификации раньше часто использовались архитектуры AlexNet (Krizhevsky et al., 2012), VGG (Simonyan, Zisserman, 2014), GoogLeNet (Szegedy et al., 2016), но после 2016 г. чаще всего используют архитектуру ResNet (Residual Networks; He et al., 2016) с различным числом слоев и ее модификации (Norouzzadeh et al., 2018; Schneider et al., 2018; Willi et al., 2019; Tabak et al., 2020; Norouzzadeh et al., 2021; van Gils, 2022; Ефремов и др., 2023б).

#### **Сверточные нейронные сети для детектирования объектов**

Задача детекции ставит своей целью локализацию объекта интереса при помощи ограничивающих рамок (Bounding Boxes, BBox), после чего определяются классы найденных объектов. Ранние CNN обходились без детектора, т.е. классифицировали все изображение целиком, не локализуя объекты интереса. Этого было достаточно, когда на изображении был объект только одного класса, но если классов несколько или требуется не только идентифицировать объекты, но и подсчитать их, необходимо их предварительное обнаружение (детектирование) (Schneider et al., 2018). Наличие ограничивающих рамок значительно повышает эффективность обнаружения объектов на изображениях (Schneider et al., 2018; Beery et al., 2018). Помимо этого увеличивается точность классификации и становится возможным производить подсчет числа особей разных видов на одном изображении (Norouzzadeh et al., 2018, 2021; Ефремов и др., 2023а, б).

Среди алгоритмов детекции выделяют двухстадийные и одностадийные. В двухстадийных детекторах на первом этапе селективным поиском или с помощью специального слоя нейросети выделяются регионы интереса (regions of interests, RoI), которые с высокой вероятностью содержат внутри себя объекты. На втором этапе такие регионы рассматриваются классификатором для определения принадлежности к исходным классам. Вместе с этим решается задача регрессии для уточнения местоположения ограничивающих рамок. Одностадийные детекторы не используют отдельный алгоритм для генерации регионов, а предсказывают координаты ограничивающих рамок, класс объекта и вероятность нахождения объектов в рамке напрямую. В качестве двухстадийных детекторов выступают R-CNN (Region-CNN; Girshick et al., 2014), Fast R-CNN (Girshick et al., 2015), Faster R-CNN (Ren et al., 2015), в то время как представителями одностадийных подходов являются SSD (Liu et al., 2016), YOLO (You Only Look Once; Redmon et al., 2016), RetinaNet (Lin et al., 2017) и т.д. Одностадийные детекторы работают значительно быстрее, чем двухстадийные, поэтому стало популярно их использовать в режиме реального времени и в системах с ограниченными вычислительными ресурсами (Feng et al., 2022). Большинство моделей, рассматриваемые далее, построены на базе одностадийных детекторов из серии YOLO (v1-v8) и двухстадийном детекторе Faster R-CNN.

Для построения и обучения нейронных сетей используются открытые библиотеки глубокого обучения на языках программирования Python и C++, например [TensorFlow](#) от компании Google, [PyTorch](#), [DarkNet](#) (Willi et al., 2019; Carl et al., 2020; Falzon et al., 2020; Tabak et al., 2020). Доступ к TensorFlow можно также получить через среду R с помощью пакета «tensorflow» (Allaire et al., 2023).

#### **Сверточные нейронные сети и фотоловушки**

Первые попытки автоматизировать процесс тегирования изображений с фотоловушек были предприняты в работе Yu et al. (2013), где авторы предложили решение на базе метода опорных векторов (support vector machine) и в Swinnen et al. (2014), где производился анализ изменения пикселей на соседних изображениях. Примерно тогда же Chen et al. (2014) предложили первую CNN с 3 сверточными и с 3 пулинговыми слоями для классификации изображений с фотоловушек Resonux. Это была совсем неглубокая по нынешним меркам нейросеть с достаточно простой архитектурой. Более продвинутое решение на основе уже глубоких CNN предложили чуть позже Gomez Villa et al. (2016, 2017), опробовав архитектуры AlexNet, VGG, GoogLeNet и ResNet для классификации датасета проекта Snapshot Serengeti (Танзания, Африка), содержащего 3.2 миллиона изображений для 48 видов животных. Тогда же ими была продемонстрирована лучшая результативность ResNet, что в дальнейшем и определило широкое ее использование. Уже через год Norouzzadeh et al. (2018) впервые опубликовали результаты многозадачного обучения, где продемонстрировали возможности не только распознавания, но и подсчета числа особей вида, а также классификации их поведения. Ограничением их работы была возможность лишь одной классификации (вид, число особей, поведение) для каждого изображения, потому что они не использовали детектор (Norouzzadeh et al., 2018). Сразу следом за ними Schneider et al. (2018) усовершенствовали данный подход, уже используя технологии детектирования объектов, обучив нейросеть определять и считать животных различных видов на одном и том же изображении.

#### **Обучение сверточных нейронных сетей**

Для корректной работы CNN необходимо их предварительное обучение на заранее подготовленных массивах данных (обучающих выборках) (Norouzzadeh et al., 2018, 2021; Vélez et al., 2023). Хорошим тоном считается создание трех наборов данных (датасетов) – тренировочного, валидационного и тестового. Большую часть данных составляет тренировочная выборка (около 70 % от исходных данных), при помощи которой подбираются параметры нейросетевой модели (весовые коэффициенты). Валидационная выборка составляет около 20 % и предназначена для оценки недообученности / переобученности модели. Также при помощи нее подбираются гиперпараметры

для алгоритма (размер пакета изображений, коэффициент обучения, количество эпох и т.д.). Тестовая выборка (около 10 % от исходных данных) предназначена для финальной оценки работоспособности модели после настройки ее параметров и гиперпараметров. Для достижения высокой оценки в качестве распознавания видов необходимо, чтобы как можно больше изображений приходилось на один класс (в идеале от 100 000; Tabak et al., 2019), причем объемы выборок для видов не должны сильно различаться (Gomez Villa et al., 2017; Norouzzadeh et al., 2018; Tabak et al., 2020), т.е. они должны быть сбалансированы (Schneider et al., 2018), а географическая представительность обучающей выборки должна отражать географический охват мест, где модель будет применяться (Beery et al., 2018; Tabak et al., 2019; Schneider et al., 2020). Также очень важно вручную проверять и контролировать результаты классификации, особенно на первых этапах, потому что даже высокие показатели точности не всегда гарантируют правильную классификацию (Guo et al., 2017; Greenberg, 2020) из-за эффекта переобучения модели. Точность хорошо обученных CNN впечатляет. Например, для такого крупного проекта, как Snapshot Serengeti, при распознавании 3.2 миллиона фотографий 48 видов животных она составила 99.3 %, тогда как результаты работы подготовленных волонтеров были лишь 96.6 % (Norouzzadeh et al., 2018).

Хорошо обученные нейросети можно использовать для поиска общих признаков на совершенно новых наборах данных с дальнейшим дообучением их находить уже специфические признаки. Это т.н. трансферное обучение (transfer learning), когда знания, полученные при решении одной задачи, применяются для решения аналогичной, но уже другой задачи (Yosinski et al., 2014). В этом случае сначала используется общая (глобальная или базовая) модель, обученная на большом (часто открытом) наборе данных (миллионы изображений), которая затем дообучается на локальном наборе данных (несколько тысяч изображений) для получения локальной (целевой) модели (Schneider et al., 2018). Технически это достигается копированием численных значений весовых коэффициентов сверточных слоев, обученных на глобальной модели, на локальную модель и переобучением у нее только весовых коэффициентов на полносвязных слоях (Willi et al., 2019). Трансферное обучение показывает свою эффективность даже на небольших датасетах.

Чем больше совпадений классов между глобальным (transfer-from) и локальным (transfer-to) наборами данных, тем лучше удается адаптировать модель (Norouzzadeh et al., 2018). Важно отметить, что модель классификатора учится на всем изображении (т.е. не только на образе самого животного, но также и на фоне за ним; Miao et al., 2019), поэтому даже при наличии одних и тех же видов, но разных фонов (например, разных локаций фотоловушек) качество классификации снизится (Willi et al., 2019; Tabak et al., 2020).

Подготовка обучающего набора данных для глобальной модели – это очень трудозатратный процесс, поэтому сейчас все больше используют алгоритмы активного обучения (active learning) (Norouzzadeh et al., 2021). В этом случае существует небольшой датасет тегированных и большой датасет нетегированных изображений, откуда лишь иногда по определенным правилам выбираются данные, которые предлагаются человеку (разметчику) для тегирования, после чего модель перестраивается (Norouzzadeh et al., 2021). Например, сначала вручную размечается небольшое число изображений (1000) и подается на вход алгоритму, после чего происходит его обучение. Затем на каждом шаге случайно выбираются неразмеченные изображения (например, по 100) и отдаются разметчику для аннотирования. После каждого шага происходит переобучение модели.

### **Автоматическая и полуавтоматическая классификации**

Распознавание образов на изображениях с фотоловушек может быть как полностью автоматическим, так и полуавтоматическим. В первом случае процесс детектирования и классификации происходит без человеческого контроля (Vélez et al., 2023). Это может быть полезно при необходимости [оперативного реагирования](#) (предотвращения браконьерства или конфликтов между человеком и крупными хищниками) в долговременных проектах с ограниченными ресурсами или в проектах без необходимости дополнительного тегирования (т.е. где нужна только видовая идентификация животных).

В тех случаях, когда точности автоматического распознавания недостаточно, применяется полуавтоматический подход. При нем компьютерное зрение совмещается с человеческим зрением, т.е. оператор выборочно проверяет результаты классификации за машиной с возможностью их корректировки (Willi et al., 2019; Vélez et al., 2023). Как правило, многие решения предоставляют возможности выбора порога уверенности классификации (confidence threshold), поэтому человеку необходимо просмотреть лишь часть изображений (ниже выбранного порога). При этом обычно происходит отсеивание пустых кадров, а кадры с животными группируются по видам. Это упрощает дальнейшую проверку результатов. В работе Willi et al. (2019) установлено, что с повышением порога точность моделей возрастает, хотя и падает охват (coverage), т.е. доля изображений с автоматически учтенной классификацией. Так, порог в 99 % давал точность видовой классификации от 96.7 до 98.9 % для разных датасетов, а охват составлял 76–86.5 % (Willi et al., 2019). Это означает, что нейросеть автоматически выполняет примерно  $\frac{3}{4}$  всей работы на надежном уровне.

### **Оценки точности сверточных нейронных сетей**

Для оценки качества работы CNN используются различные метрики в зависимости от решаемой задачи (регрессия, классификация, детекция, сегментация, трекинг и т.д.). В рамках данной работы мы рассматриваем только метрики задач классификации и детекции.

В задаче классификации зачастую используются такие метрики, как доля правильных ответов (*Accuracy*) [1], точность (*Precision*) [2], полнота (*Recall*) [3], мера ( $F_\beta$ ) [4]:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad [1]$$

$$Precision = \frac{TP}{TP + FP} \quad [2]$$

$$Recall = \frac{TP}{TP + FN} \quad [3]$$

$$F_\beta = (\beta^2 + 1) \cdot \frac{Precision \cdot Recall}{Precision + \beta^2 \cdot Recall} \quad [4]$$

В данных формулах используются следующие обозначения:

*TP* (True Positive) – количество верно предсказанных объектов положительного класса. Класс объекта считается верно предсказанным, если предсказанная положительная метка класса совпала с истинной положительной меткой класса.

*FP* (False Positive, ошибка 1-го рода) – количество ложно предсказанных объектов. Класс объекта считается ложно предсказанным, если алгоритм предсказал положительную метку класса, но объект принадлежит отрицательному классу.

*FN* (False Negative, ошибка 2-го рода) – количество ложно пропущенных объектов. Класс объекта считается ложно пропущенным, если алгоритм предсказал отрицательную метку класса, но объект имеет положительную метку класса.

*TN* (True Negative) – количество верно предсказанных объектов отрицательного класса. Класс объекта считается верно предсказанным, если предсказанная отрицательная метка класса совпала с истинной отрицательной меткой класса.

Существует подход более наглядного представления метрик *TP*, *FP*, *FN*, *TN*, используя матрицу различий (confusion matrix; рис. 2). При идеальной работе алгоритма, когда все предсказанные значения совпали с истинными, матрица примет диагональный вид, т.е. ненулевые значения будут располагаться только на главной диагонали, в противном случае недиагональные элементы будут иметь ненулевые значения.

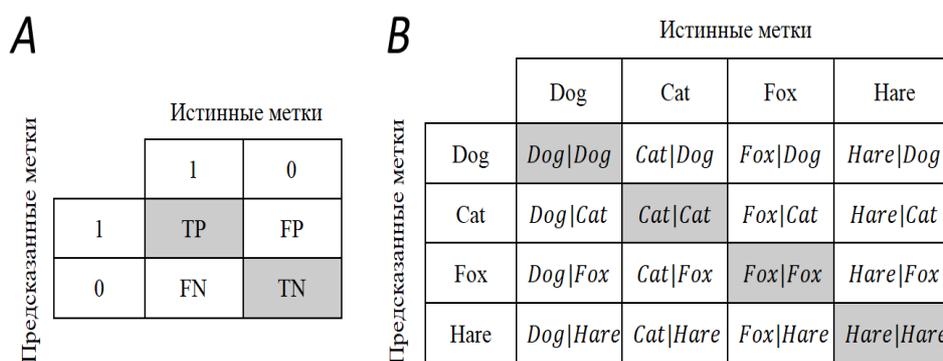


Рис. 2. Примеры матриц различий при оценке точности сверточных нейронных сетей. *A* – матрица различий для бинарной классификации. *B* – матрица различий для многоклассовой классификации на примере 4 классов (*Dog*, *Cat*, *Fox*, *Hare*). *Cat|Dog* – истинная метка не совпала с предсказанной (False Positive, *FP*). *Dog|Dog* – истинная метка совпала с предсказанной (*TP*). *Dog|Cat* – предсказанная метка не совпала с истинной (False Negative, *FN*). Серый цвет означает, что предсказание алгоритма совпало с истинным значением

Fig. 2. Examples of confusion matrices for estimating the accuracy of convolutional neural networks. *A* – confusion matrix for binary classification. *B* – confusion matrix for multiclass classification using 4 classes as an example (*Dog*, *Cat*, *Fox*, *Hare*). *Cat|Dog* – the true label did not match with the predicted label (False Positive, *FP*). *Dog|Dog* – the true label matched the predicted label (True Positive, *TP*). *Dog|Cat* is an example of False Negative (*FN*). Gray color means that the algorithm prediction matched with the true classes

В задачах, когда имеется сильный дисбаланс классов, зачастую используется  $F_\beta$  мера, которая

объединяет *Precision* и *Recall* метрики в виде гармонического среднего между ними [5]. Она устойчива к дисбалансу в отличие от *Accuracy* и может назначать приоритет между *Precision* и *Recall* при помощи коэффициента  $\beta$ . Зачастую берется равный приоритет ( $\beta = 1$ ) и формула [4] преобразуется в [5]:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

[5]

В задаче детекции, как и в задаче классификации, используются такие метрики, как *TP*, *FP*, *FN*, *Precision*, *Recall*,  $F_\beta$  мера, но также вводятся дополнительные *IoU*, *Average Precision*, *Mean Average Precision*. Метрика *IoU* показывает степень перекрытия между истинной и предсказанной ограничивающими рамками [6].

$$IoU(A, B) = \frac{A \cap B}{A \cup B}$$

[6]

Чем она больше, тем точнее алгоритм выделяет искомый объект. Значение метрики варьирует в диапазоне от 0 до 1, где 1 – это идеальное перекрытие, а 0 – отсутствие перекрытия (рис. 3).



Рис. 3. Степень перекрытия истинной (зеленый цвет) и предсказанной (красный цвет) ограничивающих рамок на примере бурого медведя (*Ursus arctos* L., 1758). А – пример *TP* ( $IoU > 0.5$ ); В – пример *FP* ( $IoU < 0.5$ ); С – пример *FP* ( $IoU = 0$ ); D – пример *FN* (пропущенный объект)

Fig. 3. Overlap degree between the true (green) and predicted (red) bounding boxes in the example of brown bear (*Ursus arctos* L., 1758). A – *TP* example ( $IoU > 0.5$ ); B – *FP* example ( $IoU < 0.5$ ); C – *FP* example ( $IoU = 0$ ); D – *FN* example (missing object)

В данном случае трактовка метрик принимает следующий вид:

*TP* – детекция объекта считается корректной, если степень перекрытия предсказанной и истинной ограничивающих рамок больше порога *IoU* ( $IoU > threshold$ );

*FP* – детекция объекта считается некорректной, если степень перекрытия предсказанной и истинной ограничивающих рамок меньше порога *IoU* ( $IoU < threshold$ );

*FN* – алгоритм не нашел объект, при этом для этого объекта существует истинная ограничивающая рамка, т.е. объект считается пропущенным;

*TN* – не применяется в задаче детекции.

Наиболее часто используемой метрикой в задаче детекции выступает *Average Precision (AP)*, которая определяется как площадь под *Precision-Recall* кривой [7]:

$$AP = \sum_{k=0}^{k=n-1} [Recalls(k) - Recalls(k+1)] \cdot Precisions(k)$$

$Recalls(n) = 0 \quad Precisions(n) = 1$

[7]

В данной формуле  $n$  – количество пороговых значений. Чем больше корректных предсказаний совершает модель, тем качественнее *Precision-Recall* кривая, следовательно, выше значение площади под этой кривой. Максимально возможное значение метрики – 1, а минимальное – 0.

В реальной жизни классов объектов может быть больше, чем один, поэтому можно посчитать *AP* метрику для каждого класса, что позволит лучше понять, на каком классе модель обрабатывает лучше всего, а на каком – хуже. В задаче мультиклассовой детекции зачастую используется метрика *mean Average Precision (mAP)*, которая усредняет значения *AP* метрики по всем классам. Метрика *mAP* считается при разных порогах *IoU*, т.к. его значение сильно влияет на конечный результат метрики. Поэтому сообществом ученых было предложено рассчитывать метрику *AP* для каждого класса и порога *IoU*, а затем усреднять полученные значения по всем классам [8]:

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k$$

[8]

В данной формуле  $n$  – количество классов. Можно встретить следующие обозначения:  $mAP@0.5$ ,  $mAP@0.5:0.95$ , которые означают значения метрики *mAP* при пороге  $IoU = 0.5$  и усредненное значение *mAP* метрики при вариации порога  $IoU$  от 0.5 до 0.95. На качественном уровне

метрика  $mAP@0.5$  показывает, насколько хорошо модель способна находить объекты, а  $mAP@0.5:0.95$  – насколько точно объекты выделяются ограничивающими рамками.

### Глобальные наборы данных для обучения сверточных нейронных сетей

С одной стороны в качестве глобальных наборов данных изображений могут использоваться такие общие источники, как ImageNet, Flickr или iNaturalist, так и наборы непосредственно изображений с фотоловушек. Основные изученные нами наборы данных представлены в табл. 1. Много других публичных датасетов изображений с фотоловушек могут быть найдены в глобальном хранилище данных (дата-репозитории) Александрийской библиотеки для биологических и природоохранных инициатив для машинного обучения Labeled Information Library of Alexandria: Biology and Conservation ([LILA BC](https://lila.science/datasets)). Отдельно стоит также отметить такой крупный международный дата-репозиторий для изображений с фотоловушек, как Wildlife Insights (Ahumada et al., 2020).

Таблица 1. Некоторые популярные глобальные наборы данных изображений для обучения сверточных нейронных сетей

Название датасета	Объем (число фотографий)	Качественный состав	URL-адрес
iNaturalist	> 45 млрд	фотографии животных и растений (также с фотоловушек)	<a href="https://www.inaturalist.org">https://www.inaturalist.org</a>
Flickr	> 10 млрд	Самые разные фотографии	<a href="https://www.flickr.com">https://www.flickr.com</a>
ImageNet (Russakovsky et al., 2015)	> 14 млн	1000 категорий от велосипедов и машин до собак и львов	<a href="https://www.image-net.org">https://www.image-net.org</a>
North America Camera Trap Image (Tabak et al., 2019)	> 3.7 млн	фотоловушки: 28 видов животных и категорий из 5 районов США	<a href="https://lila.science/datasets/nacti">https://lila.science/datasets/nacti</a>
Snapshot Serengeti (Swanson et al., 2015)	> 3.2 млн	фотоловушки: 48 видов млекопитающих и птиц Танзании (Африка)	<a href="http://www.zooniverse.org/projects/zooniverse/snapshot-serengeti">www.zooniverse.org/projects/zooniverse/snapshot-serengeti</a>
Idaho Camera Traps	> 1.5 млн	фотоловушки: 62 категории из штата Айдахо (США)	<a href="https://lila.science/datasets/idaho-camera-traps">https://lila.science/datasets/idaho-camera-traps</a>
WCS Camera Traps	> 1.4 млн	фотоловушки: 675 видов из 12 стран	<a href="https://lila.science/datasets/wcscameratraps">https://lila.science/datasets/wcscameratraps</a>
Caltech Camera Traps (Beery et al., 2018)	243 100	фотоловушки: 21 вид животных с Юго-Запада США	<a href="https://lila.science/datasets/caltech-camera-traps">https://lila.science/datasets/caltech-camera-traps</a>

### ML-программы для распознавания образов на изображениях

Далее в краткой форме приводятся описания ПО для детекции и классификации изображений. В них не предусмотрены возможности управления проектом, хранения данных, ручного тегирования, проведения анализов или построения отчетов. Обычно все они используются как сторонние и вспомогательные решения для распознавания животных на фотографиях или видео отдельно от основного ПО для обработки данных с фотоловушек.

#### MegaDetector GUI

Разработанный и поддерживаемый компаний Microsoft в рамках развития экологической инициативы «AI for Earth», MegaDetector представляет собой модель детектора, обученную на данных со всего мира, чтобы находить на изображениях с фотоловушек людей, диких животных и технику (person, animal, car, т.н. PAC-модель), а также отсортировать пустые кадры (Beery et al., 2019). В его основе лежит модель MDv5 на базе архитектуры YOLOv5, расположенная в репозитории данных компании Microsoft, откуда он может быть скачен для свободного использования. Модель MDv5 способна обрабатывать около 40 000 изображений в день на обычном компьютере или почти 1 000 000 изображений в день, используя GPU (Graphics Processing Unit) видеокарты GeForce RTX 3090. Для успешной самостоятельной работы пользователь должен хорошо разбираться в командной строке и быть готовым к написанию кода на языке Python. В качестве альтернативы возможно отправить свои данные разработчикам, которые самостоятельно их классифицируют и отправят обратно готовые результаты. На выходе MegaDetector выдает файл результатов детектирования в формате JSON, который может быть загружен в стороннее ПО, например Camelot, Zooniverse,

eMammal, digiKam или Timelapse. На сегодняшний день лучше всего осуществлена интеграция именно с Timelapse (Greenberg et al., 2019). С помощью несложного кода на Python можно просто рассортировать фотографии по папкам (пустые кадры, техника, люди, другие животные).

Для тех, кто не знаком с языком Python, существует настольное приложение [MegaDetector GUI](#) (Guurov, 2022) с дружелюбным интерфейсом, позволяющее работать с MegaDetector, не имея навыков программирования (рис. 4). На момент подготовки обзора MegaDetector стал частью более крупного проекта [Pytorch-Wildlife](#). Стоит отметить, что также имеется [MegaClassifier](#), при помощи которого можно производить более детальную классификацию по видам животных. Помимо этого возможно обучить собственный классификатор под свою фауну.

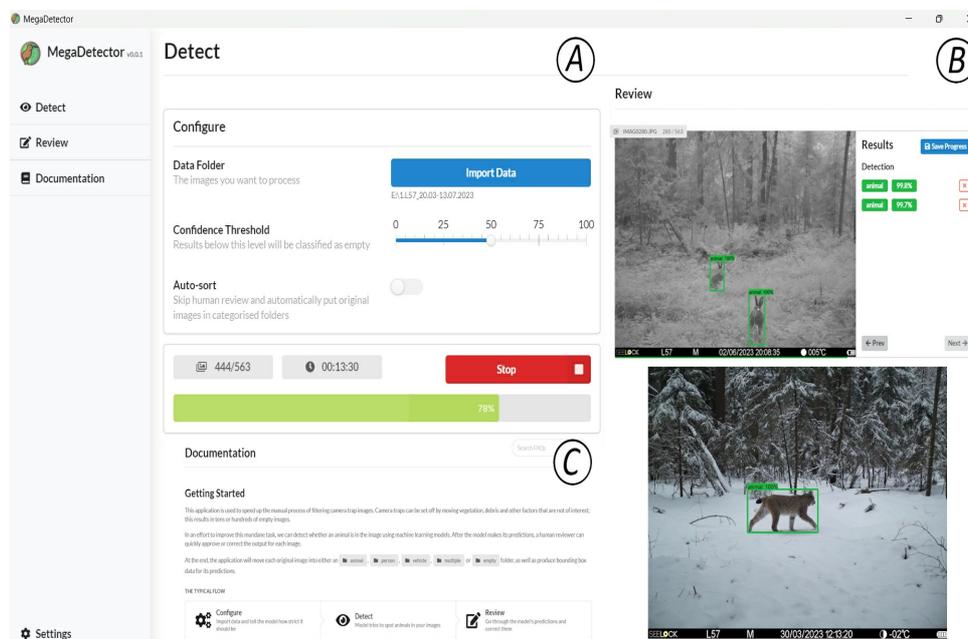


Рис. 4. Интерфейс программы MegaDetector GUI. A – окно запуска модели детектора (Detect); B – окно проверки результатов (Review); C – окно справки (Documentation)

Fig. 4. Interface of the MegaDetector GUI software. A – Detect window; B – Review window; C – Documentation window

MegaDetector используется во множестве программ и проектов по сохранению биоразнообразия и охраны природы по всему миру. Например, Департамент рыбы и дичи штата Айдахо (США) с помощью него обрабатывает данные с 2000 фотоловушек в рамках проекта по мониторингу серого волка (*Canis lupus* L., 1758). Это позволяет отсеивать значительную часть ненужных фотографий и просматривать только 15 % всех изображений. Если раньше ручное тегирование шло с запозданием на 5 лет, то теперь оно (уже полуавтоматическое) завершается еще до начала следующего сезона (Tuia et al., 2022). Согласно сравнениям, MegaDetector показал лучшие результаты относительно MLWIC2 (Vélez et al., 2023).

### **EcoAssist**

Это настольное приложение с открытым исходным кодом, предназначенное для использования ML-моделей распознавания образов на изображениях с фотоловушек. Ключевыми особенностями являются большие возможности настройки моделей, постобработки их результатов, дружелюбный интерфейс пользователя и отсутствие необходимости навыков программирования (рис. 5A; van Lunteren, 2023). В основе приложения лежит все тот же MegaDetector. Это означает, что [EcoAssist](#) предлагает в первую очередь удобные возможности применения PAC-модели: распознавания животных, людей и техники, а также отсеивания пустых снимков. В качестве детектора выступает модель YOLOv5 (van Lunteren, 2023).

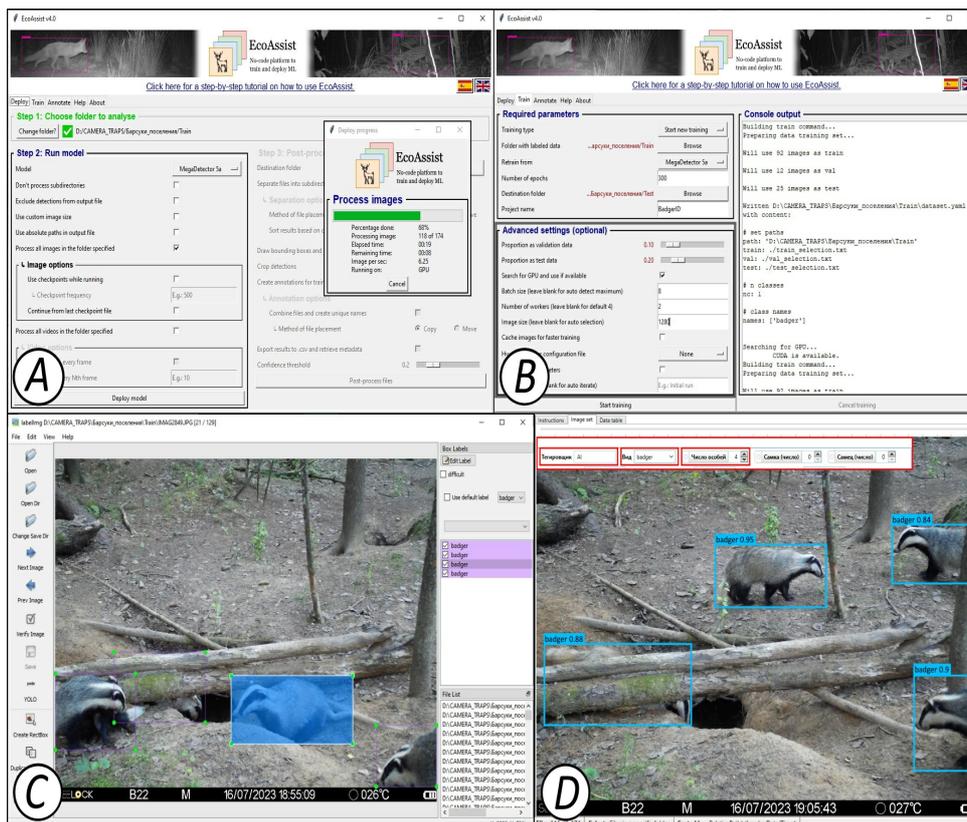


Рис. 5. Интерфейс программы EcoAssist. А – окно настроек готовой модели детектора на основе MegaDetector v5a; В – окно настроек и запуска собственной модели классификатора; С – интерфейс сторонней программы Labellingm для ручной разметки и тегирования изображений; D – часть окна сторонней программы Timelapse с интегрированными результатами модели детектора и классификатора (красной рамкой показаны результаты автозаполнения полей по результатам работы модели)

Fig. 5. Interface of the EcoAssist software. A – settings window of the ready-made detector model based on MegaDetector v5a; B – settings window and launch of the user classifier model; C – interface of the Labellingm software for manual markup and tagging of images; D – part of the window of the Timelapse software with integrated results of the detector and classifier models (the red frame shows the results of autofilling of fields based on the model results)

С помощью данного ПО возможно проводить ручную разметку, выделяя объекты ограничивающими рамками, и тегирование с помощью сторонней программы Labellingm (рис. 5С) для создания тренировочных датасетов, чтобы обучать собственные модели классификаторов (рис. 5В). Стоит отметить, что обучение своего классификатора потребует достаточных аппаратных мощностей, в частности использование GPU (предусмотрено автоматическое использование GPU для NVIDIA и Apple Silicon) (van Lunteren, 2023). Приложение работает на операционных системах Windows, Mac и Linux, не требует доступа к сети Интернет после установки, способно работать как с фотографиями, так и с видео (только формата AVI). EcoAssist полностью совместим с программой Timelapse (Greenberg et al., 2019), и его результаты могут быть интегрированы непосредственно в нее для дальнейшего тегирования (рис. 5D).

Встроенные модели детекторов MegaDetector5a и MegaDetector5b могут быть использованы для грубой обработки больших объемов данных и без обращения к GPU. С помощью них возможно детектирование и дальнейшая сортировка по папкам пустых изображений (empty), животных (animal), техники (vehicle) и людей (person) с дальнейшей постобработкой уже в программе Timelapse. Также все фотографии животных можно дополнительно отсортировать по точности детектирования.

По нашему мнению, на сегодняшний день это одно из самых продуманных и удобных открытых решений как для обученного детектора, так и для обучаемого классификатора. По результатам предварительных тестирований на основе датасета в Центральном-Лесном заповеднике точность детектора составила 98 %. Подробный обучающий ролик по EcoAssist на русском языке доступен по ссылке: <https://youtu.be/2nrXhyd-1u0>.

#### MLWIC2

Следующее программное решение называется MLWIC2 (Machine Learning for Wildlife Image Classification v. 2) и представляет собой R-пакет, разработанный специально для классификации

видов на изображениях с фотоловушек для Северной Америки. Это продолжение разработанной ранее нейросети MLWIC (Tabak et al., 2019). Помимо распознавания отдельных видов MLWIC2, можно также использовать для идентификации пустых кадров и для других географических регионов (Tabak et al., 2020). Сегодня ему доступно распознавание 58 видов млекопитающих североамериканской фауны.

С помощью MLWIC2 пользователь может самостоятельно запустить модель ИИ на своем рабочем месте без необходимости куда-то загружать или отправлять изображения. Для этого он должен установить Anaconda Navigator, Python (версии 3.5, 3.6 или 3.7), Rtools (для ОС Windows) и TensorFlow версии 1.14. Скачать MLWIC2 можно с этого [сайта](#). После работы классификатора на выходе генерируется файл, содержащий имена файлов изображений и пять лучших вариантов классификации («топ-5») для каждого из них с соответствующими порогами уверенности. Дополнительно возможно обучать свои собственные модели на заранее размеченных данных, что является главным преимуществом данного программного решения. Для этого доступны архитектуры AlexNet, DenseNet, GoogLeNet, NiN, ResNet и VGG с различным числом слоев. Лучше всех себя показала архитектура ResNet-18 с оценками точности 96.8 % для видовой классификации и 97.3 % для отсеивания пустых снимков (Tabak et al., 2020).

Поскольку нейросеть MLWIC2 строилась для фауны Северной Америки, основным ее недостатком является ограниченная применимость для других территорий. Обученный классификатор отлично показал себя для независимого датасета из Канады (точность 91 %), но продемонстрировал очень слабые результаты для датасета из штата Миссури (США; 36 %) (Tabak et al., 2020). При этом модель для распознавания пустых кадров оказалась намного более надежной. Так, для датасета Snapshot Karoo (Южная Африка) ее точность была почти 91 %, а для датасетов Snapshot Serengeti (Танзания) и Wellington (Новая Зеландия) – 94 % (Tabak et al., 2020). Программа предоставляет ограниченный список классификаторов, который можно дообучать под свое видовое разнообразие, причем эти модели значительно уступают в качестве классификации моделям 2023–2024 гг.

В отличие от некоторых других ML-программ, требующих знания языка Python, MLWIC2 использует возможности предоставления веб-интерфейса напрямую в R с помощью Shiny (Chang et al., 2019; Tabak et al., 2020), поэтому требует самые минимальные навыки программирования. Согласно результатам исследований самих разработчиков, для получения точности классификации более 95 % необходимо всего 2000 изображений для каждого вида, поэтому обучение MLWIC2 на других датасетах может иметь большие перспективы (Tabak et al., 2020). Также по оценкам разработчиков их нейросеть может классифицировать 2000 фотографий в минуту на ноутбуке с 16 Гб RAM и без обращения к графической памяти (GPU), поэтому для обучения собственной модели на базе MLWIC2 серьезные аппаратные мощности не требуются (Tabak et al., 2020). Все это открывает большие возможности для использования данной ML-программы.

### **Conservation AI**

[Conservation AI](#) является в первую очередь веб-сервисом, разработанным Ливерпульским университетом им. Джона Мура (Великобритания) совместно с NVIDIA для применения ИИ в обработке акустических данных, снимков с дронов и изображений с фотоловушек (рис. 6). На сегодняшний день оно предлагает уже готовые детекторы и классификаторы для фауны Великобритании, Южной Африки, Танзании, Северной Америки, Индомалайзийского региона и Центральной Азии. Conservation AI также предоставляет возможности для детектирования и распознавания изображений практически в реальном времени непосредственно на самих фотоловушках при их подключении через протокол SMTP (Simple Mail Transfer Protocol) и при наличии зоны покрытия сетью Интернет. Предусмотрена также обычная загрузка изображений на сайт, после чего можно выбрать нужную модель классификатора и начать распознавание, средняя скорость которого составляет примерно 10 000 изображений в час. Для доступа к сервису нужно зарегистрироваться и связаться с организаторами, чтобы активировать свой аккаунт. ПО может быть также установлено как настольное приложение на операционную систему Windows, но проще всего работать сразу через веб-браузер (желательно Google Chrome).

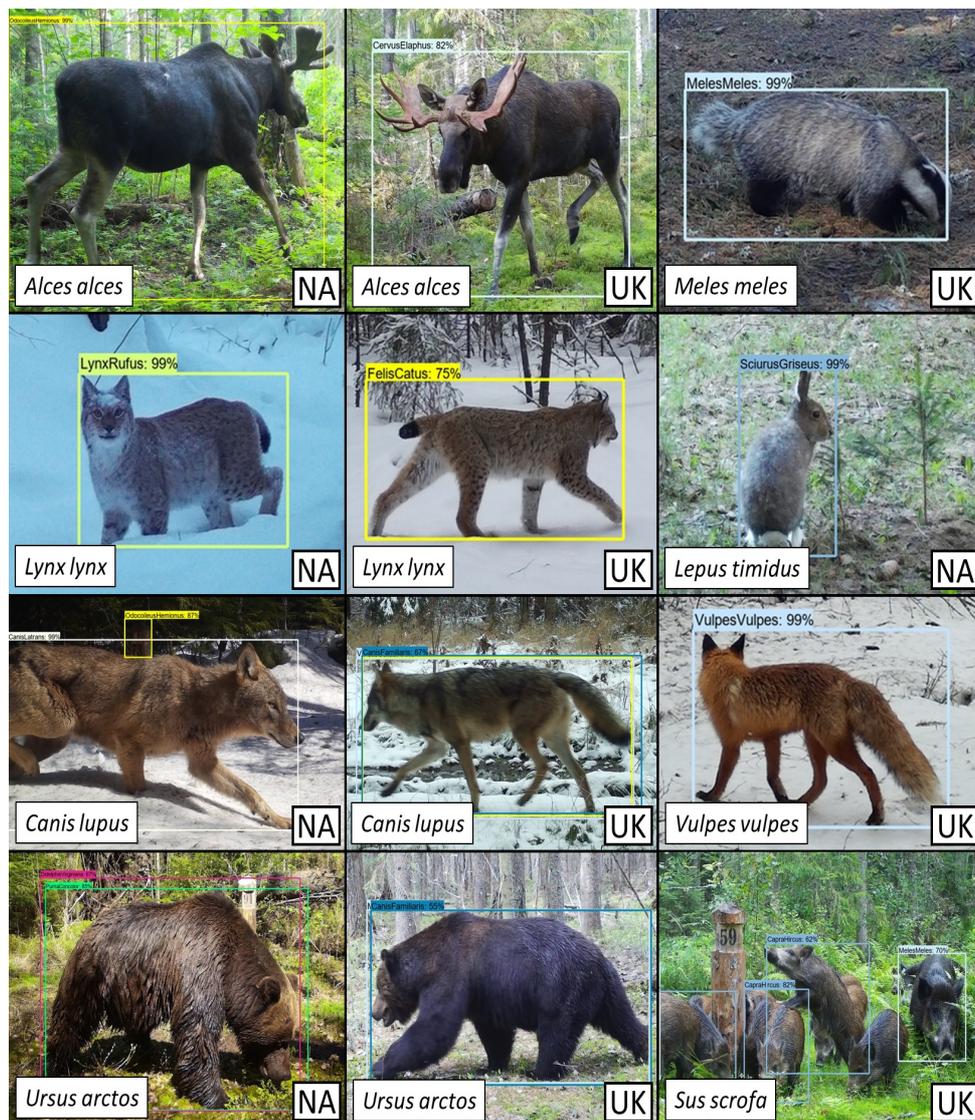


Рис. 6. Результаты классификации некоторых видов млекопитающих Центрально-Лесного заповедника (Тверская область, Россия) в программе Conservation AI. На каждом изображении внизу слева указан вид животного, внизу справа – использованная модель классификатора (NA – North American mammals; UK – United Kingdom mammals)

Fig. 6. Results of some mammal species classification from the Central Forest Nature Reserve (Tver district, Russia) in Conservation AI software. Each image shows the animal species at the bottom left and the classifier model used at the bottom right (NA – North American mammals; UK – United Kingdom mammals)

Имеющиеся модели можно улучшать, дообучая их на своих данных, или создавать свои собственные. Для этого пользователи могут проводить разметку в виде ограничивающих рамок с указанием вида животного. Для каждого вида нужно подготовить минимум 1000 изображений. Программа успешно распознает животных на фото и видео как с дронов, так и с фотоловушек. По нашему опыту, качество распознавания видов для восточноевропейской фауны (на примере ЦЛГЗ) было недостаточно хорошим даже с использованием моделей классификаторов для фаун Великобритании и Северной Америки. Также стоит отметить, что PAC-модель работает хуже, чем аналогичная у EcoAssist.

#### **FasterRCNN+InceptionResNetV2**

Помимо платформ, где представлены несколько различных моделей, можно также использовать глобальные предобученные открытые модели, например, FasterRCNN+InceptionResNetV2 (Hui, 2018). Эта модель была обучена на большом массиве данных Open Images Dataset V4 (Google LLC, 2019) и доступна для применения на сайте TensorFlow Hub (Google LLC – TensorFlow Hub 2019). Сочетание двухстадийного детектора FasterR-CNN (Ren et al., 2015) вместе с архитектурой классификатора InceptionResNetV2 позволило создать надежную модель для распознавания и классификации образов на изображениях с высокой точностью (Hui, 2018).

Загрузить изображения можно сразу же на [сайт](#). После этого необходимо выбрать потенциально-возможные виды или категории из доступного списка библиотеки Open Images V4. Для настроек и запуска модели понадобятся навыки программирования на языке Python, в частности через облачные блокноты Google Colab или Jupyter. Подробнее о настройках можно посмотреть у Huang et al. (2017) и Carl et al. (2020).

По результатам тестирования модели на случайном наборе фотографий 10 видов млекопитающих фауны Германии Carl et al. (2020) установили, что точность детектирования составила 94 %, а точность классификации на уровне отрядов – 93 %. Точность классификации на уровне вида для животных, включенных в библиотеку Open Images V4, составила 71 %. Например, класс «олень» был правильно распознан с точностью 94 %, «лисица» – 89 %, «кабан» – 83 %, «енот» – 64 %, «кошка» – 70 %, «белка» – 50 % (Carl et al., 2020). При этом на более высоких таксономических категориях (семейство, отряд, класс) точность была намного выше, доходя до 100 %. Данная модель достаточно неплохо распознает лишь общие группы животных, не проводя точную видовую классификацию.

Преимущество применения предварительно обученной модели заключается в том, что она не требует большого числа протегированных изображений, мощного компьютера и знаний в области ML. Достаточно высокая точность детектирования и высокая точность классификации для высших таксономических уровней говорят о потенциале модели (Carl et al., 2020). Она может быть использована в качестве дополнительного инструмента для анализа изображений с фотоловушек в сочетании с хорошо разработанными платформами для обработки данных, такими как Agouti (Casaer et al., 2019).

### **DeepFaune**

Это крайне простая для применения, но в то же время очень эффективная настольная программа с возможностью детектирования и классификации (Rigoudy et al., 2023). Она находится в свободном доступе, а ее дистрибутив может быть скачан с официального [сайта](#). DeepFaune является результатом сотрудничества более чем 50 различных организаций и исследовательских команд преимущественно из Франции. Изначально ее детектор основан на MegaDetector v5a (YOLO v5x), но для ускорения процесса детектирования был также разработан собственный детектор на архитектуре YOLO v8s (Rigoudy et al., 2023). По своим возможностям данное ПО похоже на EcoAssist и Conservation AI, но в отличие от них здесь имеется классификатор исключительно для европейской фауны (преимущественно Западной и Центральной Европы).

DeepFaune может быть как самостоятельной настольной программой, так и отдельным классификатором, который может быть встроен в стороннее ПО (например, он реализован в веб-сервисе Agouti). Она способна работать как с фотографиями, так и видео. После их загрузки в программу появляется возможность настроить модель классификатора, указав виды / категории животных, которых предполагается обнаружить на изображениях, а также необходимый порог уверенности и временной интервал для создания независимых регистраций (рис. 7). На сегодняшний день поддерживается 28 классов, включая категории антропогенной активности. После завершения работы модели появляется возможность автоматически подсчитать число особей на изображениях и «заблорить» изображения с людьми. Точность DeepFaune для европейской фауны очень высока. По результатам независимого тестирования ее авторов она составила 96.7 % (Rigoudy et al., 2023). В нашем случае она равняется 83 %. Такая заниженная точность обусловлена тем, что в классификаторе не оказалось двух фоновых видов млекопитающих ЦЛГЗ: европейского лося (*Alces alces* L., 1758) и енотовидной собаки (*Nyctereutes procyonoides* Gray, 1834). Также для куньих и зайцев определение идет только до ранга семейства (Mustelidae) и отряда (Lagomorpha). Если классифицировать лося как благородного оленя (*Cervus elaphus* L., 1758; каким он и определяется) с дальнейшим исправлением и оставить енотовидную собаку для ручного тегирования, то точность достигает 97 %. Результаты классификации группируются в независимые регистрации по ранее выбранному временному интервалу и записываются в CSV или XSLX-файл. Также обработанные фотографии могут быть автоматически разложены по соответствующим папкам. Единственным минусом данного ПО, препятствующим его широкому применению, является отсутствие возможности записать результаты классификации в формате JSON для дальнейшей работы с ними в других программах (например, Timelapse). Для работы с DeepFaune не требуются знания языков программирования, но для запуска должен быть предварительно установлен Python v3 с подключенным модулем PyTorch.

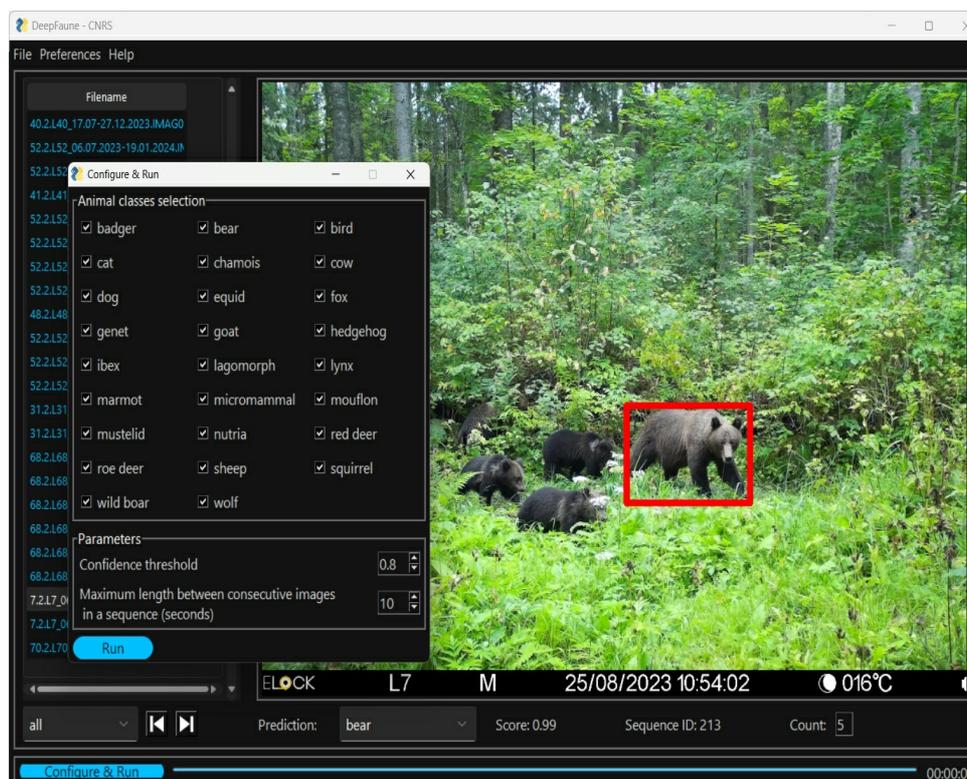


Рис. 7. Интерфейс программы DeepFaune. Показано главное окно с результатами классификации и окно настроек модели перед запуском

Fig. 7. Interface of the DeepFaune software. The main window with classification results and the model settings window before launch are shown

### **ClassifyMe**

Данное ПО разработано для полевых экологов Австралии, чтобы проводить автоматическое распознавание животных на изображениях с фотоловушек (Falzon et al., 2020). В целях сохранения информации от попадания в руки браконьеров предусмотрены регистрация и подтверждение статуса каждого пользователя через [сайт](#). Во всем остальном программа является свободной в использовании и распространении. Она разрабатывалась как настольное приложение под ОС Windows, поэтому для работы с ней не требуется постоянный доступ в сеть Интернет, лишь периодический для подтверждения своей лицензии. Дополнительные разрешения нужно также получить на отдельные модели классификаторов, которые планируется использовать, потому что модели поставляются отдельно от установочного файла и устанавливаются через соответствующие библиотеки в меню. Для классификации изображений авторы используют сверточную нейронную сеть Darknet-19 архитектуры детектора YOLOv2. Несмотря на возможности YOLOv2 детектировать и считать животных на изображениях, пока что эти функции не реализованы в ClassifyMe, но запланированы в будущем.

На сегодняшний день доступны 5 моделей классификаторов: для Австралии (точность 99 %), Новой Зеландии (точность 98 %), Танзании (точность 99 %), Северной Америки (штат Висконсин; точность 96–98 %) и Юго-Западной части США (точность 97–98.5 %) (Falzon et al., 2020). Обучение собственных моделей не предусмотрено, что делает данное ПО крайне ограниченным в использовании.

Это достаточно простая программа, предназначенная лишь для классификации и сортировки изображений с экспортом результатов в файл формата CSV. В ней не предусмотрена интегрированная база данных и функциональные возможности для менеджмента и ручного тегирования изображений. ClassifyMe автоматически распределяет изображения по новым подпапкам (субдиректориям) на основании наиболее вероятных результатов классификации с опциональной возможностью сортировки также и пустых снимков. Все результаты классификации с оценками точности для всех классов записываются в отдельный CSV-файл. ClassifyMe разрабатывался в сотрудничестве с исследователями Австралии и Новой Зеландии, а модели классификаторов обучались на их данных. Поэтому данное ПО предназначено в первую очередь для этих пользователей. Несмотря на возможности ClassifyMe работать и на других датасетах, разработчики не гарантируют надежность таких результатов (Falzon et al., 2020).

### **Программное обеспечение МФТИ**

В рамках сотрудничества с Министерством природных ресурсов и экологии Российской Федерации специалистами лаборатории систем специального назначения МФТИ была разработана

собственная программа для обработки данных с фотоловушек (Леус и др., 2023). Она построена на базе двухстадийного подхода, где на первом этапе обрабатывает детектор, задача которого состоит в нахождении объектов на фотографиях или видео. Задача второго этапа заключается в классификации объектов, найденных детектором. Для обучения нейросети использовались тренировочные выборки из 200 000 размеченных фотографий для детектора и 400 000 протегированных фотографий для обучения классификатора, собранных сотрудниками более чем 50 заповедников и национальных парков (Леус, Ефремов, 2021). Всего было задействовано 32 класса различных объектов (дикие животные, люди, техника).

В качестве модели детектора выступает одностадийный алгоритм из серии YOLO – [YOLOv5-L6](#), который был выбран как наиболее оптимальный с точки зрения скорости и качества работы по сравнению с другими алгоритмами на исходном наборе данных на момент разработки (Ефремов и др., 2023б). Классификация объектов производится с помощью нейронной сети ResNeSt101 (Zhang et al., 2020), которая сочетает в себе преимущества базового ResNet и его различных модификаций – ResNeXt, SENet, SKNet.

ПО сочетает в себе два модуля: дообучение / переобучение классификатора и автоматическая обработка данных. Модуль дообучения позволяет адаптировать модель классификатора под видовое разнообразие конкретной ООПТ при помощи технологий трансферного обучения. Модуль автоматической обработки данных использует универсальный детектор и адаптированный под конкретную ООПТ классификатор для поиска на изображениях объектов интереса и их классификации (Леус, Ефремов, 2021).

В качестве дополнительных возможностей были добавлены функции объединения изображений в независимые регистрации по установленному временному порогу, определение классов и числа особей (по максимальному значению) внутри таких регистраций (Ефремов и др., 2023а), а также автоматическая сортировка классифицированных изображений по папкам (рис. 8). ПО разрабатывалось в том числе для совместимости с отчетами программы Camelot, поэтому отдельная команда позволяет генерировать отчет в формате CSV со структурой, аналогичной SurveyExport из Camelot. Это позволяет иметь однотипный выходной файл, что упрощает дальнейший анализ данных в R.

В отличие от многих других вышеописанных решений нейросети ПО МФТИ обучались на внутриизменчивом наборе данных, полученных при совершенно разных условиях съемки, что делает возможности классификатора более универсальными и устойчивыми для применения (Shepley et al., 2021). Это стало возможным благодаря участию множества заповедников и национальных парков, которые безвозмездно поделились своими изображениями с фотоловушек.

По предварительным результатам испытаний, проведенных в Центрально-Лесном заповеднике в 2021 г. на независимых данных, точность детектирования составила более 90 %, а точность классификации – более 95 % (Леус, Ефремов, 2021). В программе пока еще не реализованы возможности хранения и менеджмента данных с фотоловушек, но уже сейчас ее можно эффективно применять для первичной обработки и сортировки большого числа изображений и подготовки их к дальнейшему ручному тегированию или непосредственно сразу к анализу. На текущий момент ПО можно приобрести по лицензии.

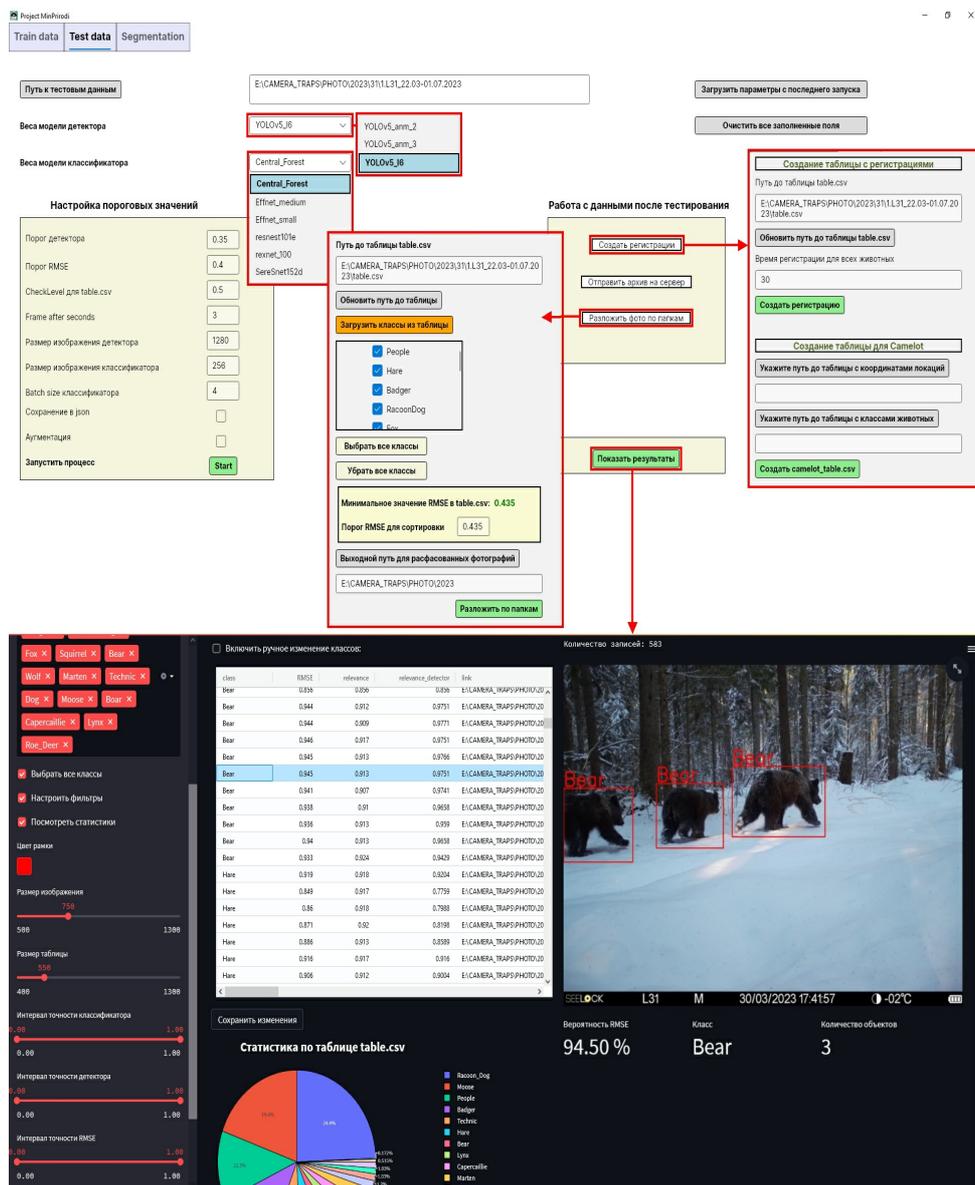


Рис. 8. Интерфейс программы МФТИ: окно настройки детектора и классификатора перед запуском (вверху) и окно просмотра результатов (внизу)

Fig. 8. Interface of the MIPT software: the window of configuring the detector and classifier before launch (top) and the window for viewing the results (bottom)

В заключение приведем сводную таблицу основных характеристик рассмотренных ML-программ (табл. 2). Из-за ограниченного объема здесь представлены лишь самые общие решения, не привязанные исключительно к конкретным проектам. Поскольку рассматриваемые программы обладают разными возможностями как в области детекции, так и в области классификации и используют для этих задач разные модели, мы не приводим единых количественных оценок их работы.

Таблица 2. Сравнительные характеристики представленных ML-сервисов и программ

Название программы	MegaDetector	EcoAssist	MLWIC2	Conservation AI	DeepFaune	ClassifyMe	ПО МФТИ
Тип	Веб-сервис, десктоп	Десктоп	R-пакет	Веб-сервис, десктоп	Десктоп	Десктоп	Десктоп
Детектор	Есть	Есть	Нет	Есть	Есть	Есть	Есть
Архитектура детектора	YOLOv5	YOLOv5	-	Faster R-CNN	YOLOv5, YOLOv8	YOLOv2	YOLOv5

Классификатор установленный	Нет	Нет	США	Англия, Южная Африка, Танзания, Северная Америка и др.	Европа (кроме Северной и Восточной)	Австралия, Новая Зеландия, Танзания, США	ООПТ России
Классификатор пользователя	Нет	Есть	Есть	Есть	Нет	Нет	Есть
Архитектура классификатора	-	-	AlexNet, DenseNet, GoogLeNet, NiN, ResNet, VGG	ResNet-101	ConvNext-Base	DarkNet-19	ResNet, EffNet, RexNet, SereSnet, ResNeSt
Графический интерфейс	Есть (MegaDetector GUI)	Есть	Есть (Shiny)	Есть	Есть	Есть	Есть
Навыки	Python	Не нужны	Базовые R	Не нужны	Не нужны	Не нужны	Не нужны
Использование	Открыто	Открыто	Открыто	Открыто	Открыто	Ограничено	Для ООПТ

### Прочие решения

К сожалению, мы не смогли охватить в одном обзоре все имеющиеся программы, потому что сегодня их уже достаточно много, и их число постоянно растет (подробнее см. [сайт](#) Дэна Морриса). Кроме описанного ПО существует ряд других программ для распознавания образов на изображениях с фотоловушек, среди которых AnimalFinder – детектор с доступом через MATLAB (Tack et al., 2016); Animal Scanner (Yousif et al., 2019) – также детектор (с классификацией на пустые кадры, людей и диких животных) с доступом как через MATLAB GUI, так и командную строку; детектор CamTrap-detector (Evans, 2023); открытая DNN, разработанная для проекта [Snapshot Serengeti](#) (Norouzzadeh et al., 2018); Zilong – программа, созданная для автоматического распознавания пустых изображений с фотоловушек без ML (Wei et al., 2020). Особого внимания заслуживают веб-сервисы AIDE (Annotation Interface for Data-driven Ecology) с технологией активного обучения, предоставляющий широкие возможности для ручной и автоматической классификации животных на изображениях как с фотоловушек, так и с дронов (Kellenberger et al., 2020), и [WildBook](#), объединяющий систематические исследования с фотоловушками, дронами и гражданскую науку с самыми последними достижениями в области машинного обучения для идентификации видов, особей и расчета популяционных характеристик для более чем 50 видов по всему миру (Berger-Wolf et al., 2017).

### Обсуждение

Одним из самых трудозатратных этапов при обработке данных с фотоловушек является процесс их аннотирования / тегирования, т.е. классификация изображений и присвоение им дополнительной информации (Reyserhove et al., 2023). Так, недавний глобальный опрос выявил, что 61 % исследователей считают обработку и анализ изображений существенным препятствием для эффективных исследований с фотоловушками (Glover-Kapfer et al., 2019). Технологии ИИ могут значительно упростить тегирование, сэкономив таким образом много времени (Norouzzadeh et al., 2018, 2021; Schneider et al., 2019; Vélez et al., 2023). Согласно Norouzzadeh et al. (2018), в проекте Snapshot Serengeti для того, чтобы вручную протегировать (вид, число особей, детеныши, поведение) примерно 5.5 миллионов кадров, используя около 30 000 волонтеров, понадобится работать полную рабочую неделю (40 часов) в течение 14.6 года. В то же время разработанная ими DNN сэкономила около 8.4 года (почти 17 500 часов) такой работы, протегировав почти 3.2 миллиона изображений.

Несмотря на то, что для обучения собственных моделей необходимы очень большие объемы данных (сотни тысяч или даже миллионы изображений), применение ИИ доступно не только для крупных проектов. Небольшие исследования также могут применять ML-модели, используя уже описанные трансферное обучение и глобальные наборы данных. Norouzzadeh et al. (2018) провели расчеты и выяснили, что улучшение дообучения локальной модели повышается с загрузкой все большего числа изображений. Эти авторы установили, что с начальным набором из 1500 фотографий можно автоматически протегировать 41 % с заданной точностью 96.6 % (точность работы обученных волонтеров в проекте Snapshot Serengeti). При условии просмотра каждой фотографии в течении 10 с тегирование такого объема займет 4.2 часа. Если обучить модель уже на 3000 фотографиях (8.3 часа ручного тегирования), автоматизировать рабочий процесс можно более чем на 50 %. С появлением 6, 10 и 15 тысяч изображений (16.7, 27.8 и 41.7 ч.) автоматизация

составляет 62.6, 71.4 и 83.0 %, а при 50 000 фотографий (138.9 ч.) 91.4 % работы может быть целиком выполнено ИИ (Norouzzadeh et al., 2018). Willi et al. (2019) выяснили, что даже на небольшом наборе данных (17 671 изображений) точность модели на основе трансферного обучения составила 85.8 %.

Согласно выводам недавнего обзора Vélez et al. (2023), многие популярные программы с применением ИИ (Conservation AI, MLWIC2, Wildlife Insights) дают низкие оценки точности распознавания при обработке данных со сторонних локаций, независимо от их количества, а также таксономического и географического разнообразия (см. рис. 6). Если на уровне семейств точность (*Precision*) классификатора достаточно надежна (более 90 %), то на видовую диагностику до сих пор положиться нельзя (оценки *Recall* менее 70 %) (Vélez et al., 2023). Это подтвердило прежние выводы о том, что ИИ-модели все еще плохо работают на данных с новых локалитетов (Schneider et al., 2020; Tabak et al., 2020) и результаты их классификации сильно разнятся для разных видов (Whytock et al., 2021). В то же время, если подобрать оптимальный сторонний классификатор (например, DeepFaune для ЦЛГЗ), то точность может быть поразительно высока. Она не будет доходить до точности собственных глобальных моделей (например, из ПО МФТИ), но будет довольно близка к ней.

Даже в рамках проекта, для которого была обучена модель, необходимо постоянно дообучать ее классификатор, потому что точность распознавания будет ниже на каждом новом массиве данных (например, за следующие года) даже с прежних локаций (Norouzzadeh et al., 2021). В этом случае перспективным направлением является активное обучение, где оператор тегирует лишь часть изображений, в которых машина не уверена, а затем они отправляются на дообучение и процесс повторяется заново (Sener, Savarese, 2018). Также важно соблюдать равномерные выборки достаточного объема для обучения моделей, в том числе для RGB и IR (InfraRed) изображений (Tuia et al., 2022).

На сегодняшний день принято считать, что наиболее эффективное применение ИИ в исследованиях с фотоловушками заключается в трех основных сферах использования: 1) автоматическое отсеивание пустых изображений (например, в случае «шевеленки»); 2) автоматическое распознавание видов лишь при очень высоком значении (например, больше 0.95–0.98) проходного порога уверенности; 3) предоставление пользователю «топ-5» лучших результатов классификации для его экспертного выбора правильного варианта (Norouzzadeh et al., 2018; Glover-Karfer et al., 2019; Green et al., 2020; Vélez et al., 2023). Полностью автоматическое распознавание пока что возможно лишь в крупных продолжительных проектах, на данных которых были обучены собственные глобальные модели (Green et al., 2020). Для всех остальных случаев целесообразнее применять полуавтоматическую классификацию или использовать ИИ лишь для построения PAC-модели. Современные ML-платформы могут эффективно использоваться для упрощения процесса тегирования путем предварительного поиска животных на фотографиях и их выделения ограничивающими рамками (с помощью детекторов) с разбиением на грубые классы (пустые кадры, дикие животные, люди, техника). В дальнейшем это значительно упрощает идентификацию до видового уровня, используя уже ручное тегирование или подходящие CNN-классификаторы (Beery et al., 2021). Например, Fennel et al. (2022) установили, что использование MegaDetector повышает производительность тегирования на 500 % по сравнению с исключительно ручной обработкой.

Таким образом, на сегодняшний день наиболее эффективным остается полуавтоматическая классификация, когда пользователь может настраивать порог уверенности и затем вручную проверять лишь часть результатов. При этом стоит отметить, что, понижая порог, мы завышаем оценки *Recall*, потому что сокращаем долю пропущенных животных, но также и занижаем оценки *Precision*, т.е. увеличиваем долю ложноположительных классификаций. Это потребует большей вовлеченности оператора в проверку результатов классификации (Vélez et al., 2023). Для каждого отдельного проекта необходимо подбирать свои пороговые значения.

Исходя из этого, среди всех рассмотренных программ наиболее подходящими для широкой аудитории мы считаем EcoAssist и MegaDetector GUI. Оба этих решения являются свободными для использования и не требуют навыков программирования. Они одинаково хорошо подходят для детектирования, грубой классификации по PAC-модели и сортировки снимков. Оба ПО имеют полную синхронизацию со сторонней программой Timelapse, предназначенной для ручного тегирования изображений (подробнее см.: Огурцов и др., 2024). В то же время мы рекомендуем использовать именно EcoAssist ввиду ее большей скорости обработки изображений, более продвинутых возможностей по сравнению с MegaDetector GUI и более удобному интерфейсу. Conservation AI в целом также может быть использована, но ее точность распознавания хуже, чем у вышеописанных программ. Для исследователей, работающих с европейской фауной, лучшим решением для классификации будет DeepFaune.

Пользователям, заинтересованным в полностью автоматической классификации, стоит рассмотреть возможности создания собственных моделей (например, с помощью EcoAssist, MLWIC2 или Conservation AI), но это потребует значительных усилий по созданию обучающих наборов данных. Также возможно использовать глобальные уже предобученные открытые модели (например, FasterRCNN+InceptionResNetV2), дообучив их на своих данных. Представителям

российских ООПТ повезло гораздо больше, потому что модели, разработанные МФТИ, уже натренированы для большинства видов фауны РФ и различных условий съемки.

Несмотря на огромные возможности, которые открывает перед нами глубокое машинное обучение, на наш взгляд, следует очень осторожно к этому относиться при обработке данных с фотоловушек. Автоматическая классификация может быть оправдана лишь в действительно глобальных проектах с сотнями камер либо при необходимости принятия оперативных решений. Во всех остальных случаях (при наличии менее 100 фотоловушек) лучшим способом остается полуавтоматическое тегирование с применением RAS-модели.

Также не стоит забывать, что помимо основных данных (вид животного и число особей) изображения с фотоловушек содержат массу другой биологической и экологической информации, которую ИИ оценить пока еще не в состоянии. Помимо половозрастной характеристики и фенологических явлений (например, линьки или роста рогов), это могут быть интересные особенности поведения, межвидовые взаимодействия, фенотипические особенности особей, их физиологическое состояние и заболевания (рис. 9). К фотоловушкам следует относиться как к инструменту для получения научных данных, но не стоит забывать, что это в первую очередь изображения, которые пока еще нужно просматривать человеку.

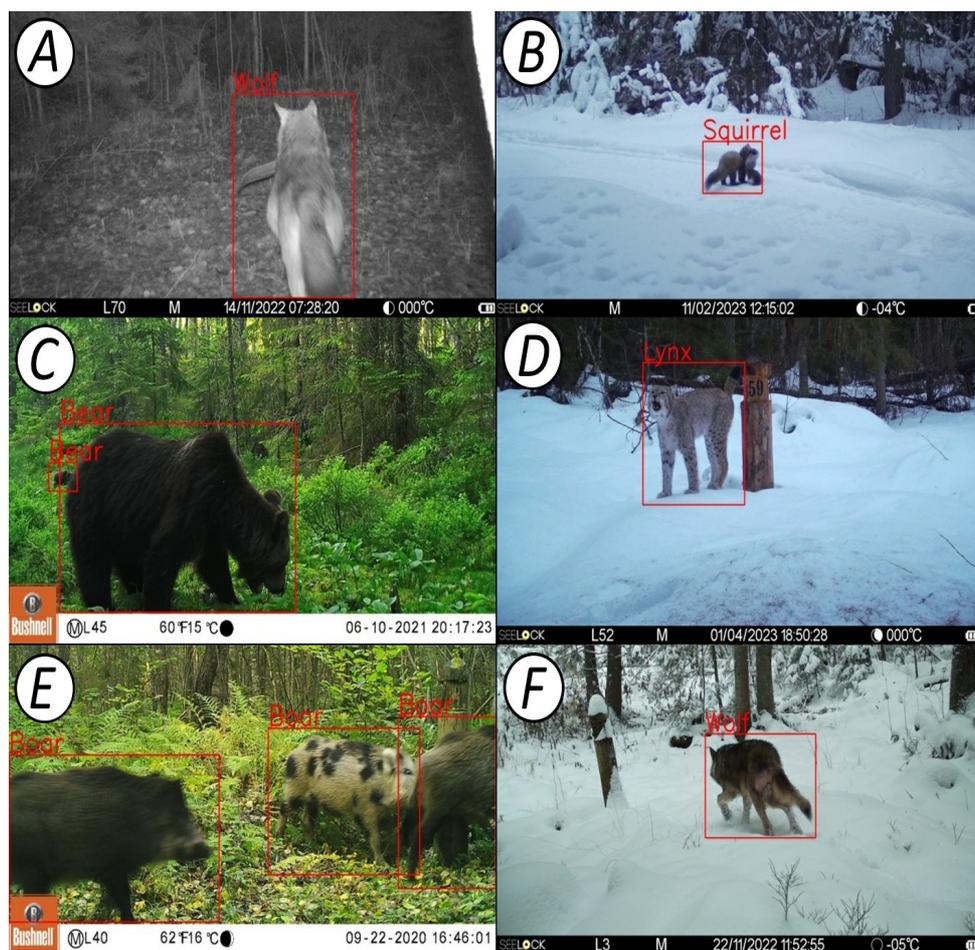


Рис. 9. Примеры важных биологических и экологических наблюдений, полученных по кадрам с фотоловушек, пока недоступных для автоматического анализа ИИ, с результатами классификации ПО МФТИ. А – волк с добытым бобром (*Castor fiber* L., 1758); В – лесная куница (*Martes martes* L., 1758) перетаскивает белку (*Sciurus vulgaris* L., 1758) из своей заначки; С – семья бурых медведей питается побегами вахты трехлистной (*Menyanthes trifoliata* L., 1753); D – вокализация самца рыси (*Lynx lynx* L., 1758) в период позднего гона; E – пятнистая окраска (фенотипическая морфа) кабана (*Sus scrofa* L., 1758); F – заболевание волка (возможно, чесотка или стригущий лишай)

Fig. 9. Examples of important biological and ecological observations obtained from camera trap images, not yet available for automatic AI analysis, with classification results from MIPT software. A – grey wolf with a preyed beaver (*Castor fiber* L., 1758); B – pine marten (*Martes martes* L., 1758) dragging a squirrel (*Sciurus vulgaris* L., 1758); C – a family of brown bears feeding on the shoots of *Menyanthes trifoliata* L. (1753); D – vocalization of a lynx (*Lynx lynx* L., 1758) male during the late mating season; E – spotted coloration of wild boar (*Sus scrofa* L., 1758); F – wolf's disease (probably scabies or ringworm)

## Заключение

Область применения ИИ в распознавании образов животных на изображениях с фотоловушек стремительно развивается и еще только начинает свое становление. Модели детекторов и классификаторов непрерывно учатся, а объемы обучающих наборов данных постоянно растут. Это позволяет делать оптимистичные прогнозы уже на ближайшее будущее, где ИИ будет способен не только успешно распознавать виды, но и половозрастные характеристики, а также поведение, возраст особей и их самих (Tuia et al., 2022; van Gils, 2022; Shi et al., 2023). Важно отметить, что качественный прогресс в этой области возможен только при тесном сотрудничестве экологов и природоохранных биологов с программистами и ML-специалистами. Мы полностью согласны с Tuia et al. (2022), что оба этих больших научных сообщества должны работать вместе, чтобы разрабатывать новые инструменты, анализы и подходы для сохранения биологического разнообразия на нашей планете.

В то же время не следует гнаться за применением ИИ в экологии как за самоцелью. Необходимо очень тщательно и взвешенно подходить к результатам любых ML-моделей и учитывать потенциальные риски при использовании их результатов, потому что плата за ошибки может быть очень высока (Tuia et al., 2022). Приоритетом должны всегда оставаться охрана и изучение дикой природы, а все принимаемые решения следует тщательно взвешивать на предмет возможных негативных последствий для нее. В данном случае ИИ должен выступать как сторонний помощник только там, где он действительно необходим.

## Библиография

- Белявский Д. С. Применение искусственных нейронных сетей при исследовании популяций животных // Охрана окружающей среды и заповедное дело. 2022. № 3. С. 81–88.
- Бизиков В. А., Сабиров М. А., Сидоров Л. К., Лукина Ю. Н. Численность и распределение ладожской кольчатой нерпы в аномально теплую зиму 2020 года: оценка по результатам авиаучета с использованием БПЛА // Труды ВНИРО. 2022. Т. 190. С. 79–94. DOI: 10.36038/2307-3497-2022-190-79-94
- Ефремов В. А., Зуев В. А., Леус А. В., Мангазеев Д. И., Радыш А. С., Холодняк И. В. Формирование регистраций животных на основе постобработки // Экосистемы. 2023а. Вып. 34. С. 51–58.
- Ефремов В. А., Леус А. В., Гаврилов Д. А., Мангазеев Д. И., Холодняк И. В., Радыш А. С., Зуев В. А., Водичев Н. А. Метод обработки фото- и видеоданных с фотоловушек с использованием двухстадийного нейросетевого подхода // Искусственный интеллект и принятие решений. 2023б. № 3. С. 98–108. DOI: 10.14357/20718594230310
- Леус А. В., Гаврилов Д. А., Мангазеев Д. И., Ефремов В. А., Радыш А. С., Зуев В. А., Холодняк И. В. Система анализа данных, считываемых с помощью фотоловушек, для оперативного дистанционного мониторинга природных территорий. Патент Российской Федерации №2799114. Федеральный институт промышленной собственности, 2023.
- Леус А. В., Ефремов В. А. Применение методов компьютерного зрения для анализа изображений, собранных с фотоловушек в рамках программно-аппаратного комплекса мониторинга состояния окружающей среды на особо охраняемых природных территориях // Труды Мордовского государственного природного заповедника им. П. Г. Смидовича. 2021. Вып. 28. С. 121–129.
- Михайлов В. В., Колпащиков Л. А., Соболевский В. А., Соловьев Н. В., Якушев Г. К. Методологические подходы и алгоритмы распознавания и подсчета животных на аэрофотоснимках // Информационно-управляющие системы. 2021. № 5. С. 20–32. DOI: 10.31799/1684-8853-2021-5-20-32
- Огурцов С. С., Ефремов В. А., Леус А. В. Обзор программного обеспечения для обработки и анализа данных с фотоловушек: нейронные сети и веб-сервисы // Russian Journal of Ecosystem Ecology. 2024.
- Ahumada J. A., Fegraus E., Birch T., Flores N., Kays R., O'Brien T. G., Palmer J., Schuttler S., Zhao J. Y., Jetz W., Kinnaird M., Kulkarni S., Lyet A., Thau D., Duong M., Oliver R., Dancer A. Wildlife Insights: A Platform to Maximize the Potential of Camera Trap and Other Passive Sensor Wildlife Data for the Planet // Environmental Conservation. 2020. Vol. 47. P. 1–6. DOI: 10.1017/S0376892919000298
- Allaire J. J., Kalinowski T., Falbel D., Eddelbuettel D., Tang Y., Golding N. Package “tensorflow”: R Interface to TensorFlow. R package version 2.14.0. 2023. URL: <https://cran.r-project.org/package=tensorflow> (дата обращения: 25.10.2023).
- Beery S., Van Horn G., Perona P. Recognition in Terra Incognita // Proceedings of the European Conference on Computer Vision (ECCV) / V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.). Munich, Germany: Springer, 2018. P. 456–473.
- Beery S., Morris D., Yang S. Efficient pipeline for camera trap image review // arXiv. 2019. Article: 1907.06772. DOI: 10.48550/arXiv.1907.06772
- Beery S., Agarwal A., Cole E., Birodkar V. The iWildCam 2021 Competition Dataset // arXiv. 2021. DOI: 10.48550/arXiv.2105.03494
- Berger-Wolf T. Y., Rubenstein D. I., Stewart C. V., Holmberg J. A., Parham J., Menon S., Crall J., Van Oast J., Kiciman E., Joppa L. Wildbook: Crowdsourcing, computer vision, and data science for conservation // arXiv. 2017. Article: 1710.08880. DOI: 10.48550/arXiv.1710.08880

- Binta Islam S., Valles D., Hibbitts T. J., Ryberg W. A., Walkup D. K., Forstner M. R. J. Animal Species Recognition with Deep Convolutional Neural Networks from Ecological Camera Trap Images // *Animals*. 2023. Vol. 13. Article: 1526. DOI: 10.3390/ani13091526
- Bogucki R., Cygan M., Khan C. B., Klimek M., Milczek J. K., Mucha M. Applying deep learning to right whale photo identification // *Conservation Biology*. 2018. Vol. 33, No 3. P. 676–684. DOI: 10.1111/cobi.13226
- Bridle J. S. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition // *Neurocomputing*. New York: Springer, 1990. P. 227–236.
- Carl C., Schönfeld F., Profft I., Klamm A., Landgraf D. Automated detection of European wild mammal species in camera trap images with an existing and pretrained computer vision model // *European Journal of Wildlife Research*. 2020. Vol. 66, No 4. P. 1–7. DOI: 10.1007/s10344-020-01404-y
- Casaer J., Milotic T., Liefing Y., Desmet P., Jansen P. Agouti: A platform for processing and archiving camera trap images // *Biodiversity Information Science and Standards*. 2019. Vol. 3. Article: e46690. DOI: 10.3897/biss.3.46690
- Ceballos G., Ehrlich P. R., Raven P. H. Vertebrates on the brink as indicators of biological annihilation and the sixth mass extinction // *Proceedings of the National Academy of Science*. 2020. Vol. 117, No 24. P. 13596–13602. DOI: 10.1073/pnas.1922686117
- Chalmers C., Fergus P., Wich S., Montanez A. C. Conservation AI: Live stream analysis for the detection of endangered species using convolutional neural networks and drone technology // *arXiv*. 2019. Article: 1910.07360. DOI: 10.48550/arXiv.1910.07360
- Chang W., Cheng J., Alaire J., Xie Y., McPherson J. Shiny: Web application framework for R. R package version 1.4.0. 2019. URL: <https://CRAN.R-project.org/package=shiny> (дата обращения: 25.10.2023).
- Chen G., Han T. X., He Z., Kays R., Forrester T. Deep convolutional neural network based species recognition for wild animal monitoring // *Proceedings of the IEEE International Conference on Image Processing*. Paris: IEEE, 2014. P. 858–862. DOI: 10.1109/ICIP.2014.7025172
- Chen P., Swarup P., Matkowski W. M., Kong A. W. K., Han S., Zhang Z., Rong H. A study on giant panda recognition based on images of a large proportion of captive pandas // *Ecology and Evolution*. 2020. Vol. 10, No 7. P. 3561–3573. DOI: 10.1002/ece3.6152
- Corcoran E., Winsen M., Sudholz A., Hamilton G. Automated detection of wildlife using drones: Synthesis, opportunities and constraints // *Methods in Ecology and Evolution*. 2021. Vol. 12, No 6. P. 1103ionndr. DOI: 10.1111/2041-210X.13581
- Evans B. C. CamTrap-detector: Detect animals, humans and vehicles in camera trap imagery. 2023. URL: <https://github.com/bencevans/camtrap-detector> (дата обращения: 25.10.2023).
- Falzon G., Lawson C., Cheung K.-W., Vernes K., Ballard G. A., Fleming P. J. S., Glen A. S., Milne H., Mather-Zardain A. T., Meek P. D. ClassifyMe: A field-scouting software for the identification of wildlife in camera trap // *Animals*. 2020. Vol. 10, No 1. Article: 58. DOI: 10.3390/ani10010058
- Farley S. S., Dawson A., Goring S. J., Williams J. W. Situating ecology as a big-data science: current advances, challenges, and solutions // *BioScience*. 2018. Vol. 68. P. 563–576. DOI: 10.1093/biosci/biy068
- Feng H., Mu. G., Zhong S., Zhang P., Yuan T. Benchmark Analysis of YOLO Performance on Edge Intelligence Devices // *Cryptography*. 2022. Vol. 6, No 2. Article: 16. DOI: 10.3390/cryptography6020016
- Fennell M., Beirne C., Burton A. C. Use of object detection in camera trap image identification: Assessing a method to rapidly and accurately classify human and animal detections for research and application in recreation ecology // *Global Ecology and Conservation*. 2022. Vol. 35. Article: e02104. DOI: 10.1016/j.gecco.2022. e02104
- Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, USA: IEEE, 2014. P. 580–587. DOI: 10.1109/CVPR.2014.81
- Girshick R. Fast R-CNN // *Proceedings of the IEEE International Conference on Computer Vision*. Santiago, Chile: IEEE, 2015. P. 1440–1448. DOI: 10.1109/ICCV.2015.169
- Glover-Kapfer P., Soto-Navarro C. A., Wearn O. R. Camera-trapping version 3.0: Current constraints and future priorities for development // *Remote Sensing in Ecology and Conservation*. 2019. Vol. 5. P. 209–223. DOI: 10.1002/rse2.106
- Gomez Villa A., Diez G., Salazar A., Diaz A. Animal identification in low quality camera-trap images using very deep convolutional neural networks and confidence thresholds // *International Symposium on Visual Computing*. Cham, Switzerland: Springer, 2016. P. 747–756. DOI: 10.1007/978-3-319-50835-1\_67
- Gomez Villa A., Salazar A., Vargas F. Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks // *Ecological Informatics*. 2017. Vol. 41. P. 24–32. DOI: 10.1016/j.ecoinf.2017.07.004.
- Goodfellow I., Bengio Y., Courville A. *Deep Learning*. Cambridge, USA: MIT Press, 2016. 800 p.
- Google LLC - TensorFlow Hub. faster\_rcnn/inception\_resnet\_v2. 2019. URL: [https://tfhub.dev/google/faster\\_rcnn/openimages\\_v4/inception\\_resnet\\_v2/1](https://tfhub.dev/google/faster_rcnn/openimages_v4/inception_resnet_v2/1) (дата обращения: 27.10.2023).
- Google LLC. Open Images Dataset V4, under CC BY 4.0 license. 2019. URL: [https://storage.googleapis.com/openimages/web/factsfigures\\_v4.html](https://storage.googleapis.com/openimages/web/factsfigures_v4.html) (дата обращения: 27.10.2023).

- Green S. E., Rees J. P., Stephens P. A., Hill R. A., Giordano A. J. Innovations in camera trapping technology and approaches: The integration of citizen science and artificial intelligence // *Animals*. 2020. Vol. 10, No 1. Article: 132. DOI: 10.3390/ani10010132
- Greenberg S. Automated image recognition for wildlife camera traps: making it work for you. Technical report. 2020. URL: <http://hdl.handle.net/1880/112416> (дата обращения: 22.03.2023).
- Greenberg S., Godin T., Whittington J. Design patterns for wildlife-related camera trap image analysis // *Ecology and Evolution*. 2019. Vol. 9, No 24. P. 13706–13730. DOI: 10.1002/ece3.5767
- Guo C., Pleiss G., Sun Y., Weinberger K. On calibration of modern neural networks // *arXiv*. 2017. Article: 1706.04599. DOI: 10.48550/arXiv.1706.04599
- Gyurov P. MegaDetector-GUI: A desktop application that makes using MegaDetector's model easier. 2022. URL: <https://github.com/petargyurov/megadetector-gui> (дата обращения: 21.10.2023).
- Hagan M. T., Demuth H. B., Beale M. H., De Jesús O. *Neural network design*. Boston: Pws Publications Co, 1996. 1011 p.
- He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition // *Proceedings of the IEEE conference on computer vision and pattern recognition*. Las Vegas, USA: IEEE, 2016. P. 770–778. DOI: 10.1109/CVPR.2016.90
- Hu W., Huang Y., Wei L., Zhang F., Li H. Deep convolutional neural networks for hyperspectral image classification // *Journal of Sensors*. 2015. Vol. 2. P. 1–12. DOI: 10.1155/2015/258619
- Hui J. Object detection: speed and accuracy comparison (faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3). 2018. URL: [https://medium.com/@jonathan\\_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359](https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359) (дата обращения: 22.10.2023).
- Huang J., Rathod V., Sun C., Zhu M., Korattikara A., Fathi A., Murphy K. Speed/accuracy trade-offs for modern convolutional object detectors // *Proceedings of the IEEE conference on computer vision and pattern recognition*. Honolulu, USA: IEEE, 2017. P. 3296–3297. DOI: 10.1109/CVPR.2017.351
- Kellenberger B., Veen T., Folmer E., Tuia D. 21 000 birds in 4.5 h: efficient large-scale seabird detection with machine learning // *Remote Sensing in Ecology and Conservation*. 2021. Vol. 7, No 3. P. 445–460. DOI: 10.1002/rse2.200
- Kellenberger B., Tuia D., Morris D. AIDE: Accelerating image-based ecological surveys with interactive machine learning // *Methods in Ecology and Evolution*. 2020. Vol. 11, No 12. P. 1716–1727. DOI: 10.1111/2041-210X.13489
- Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks // *Advances in Neural Information Processing Systems*. La Jolla, USA: Neural Information Processing Systems Foundation, 2012. P. 1097–1105. DOI: 10.1145/3065386
- Kwok R. AI empowers conservation biology // *Nature*. 2019. Vol. 567. P. 133–135. DOI: 10.1038/d41586-019-00746-1
- LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D. Backpropagation applied to handwritten zip code recognition // *Neural Computation*. 1989. Vol. 1, No 4. P. 541–551. DOI: 10.1162/neco.1989.1.4.541
- LeCun Y., Bengio Y., Hinton G. Deep learning // *Nature*. 2015. Vol. 521. P. 436–444. DOI: 10.1038/nature14539
- Lin T.-Y., Goyal P., Girshick R., He K., Dollár P. Focal loss for dense object detection // *Proceedings of the IEEE International Conference on Computer Vision*. Venice, Italy: IEEE, 2017. P. 2999–3007. DOI: 10.1109/ICCV.2017.324
- Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A. C. SSD: single shot multibox detector // *European Conference on Computer Vision / B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.)*. Amsterdam: Springer, 2016. P. 21–37. DOI: 10.48550/arXiv.1512.02325
- Meek P. D., Ballard G. A., Falzon G., Williamson J., Milne H., Farrell R., Stover J., Mather-Zardain A. T., Bishop J., Cheung E. K.-W., Lawson C. K., Munezero A. M., Schneider D., Johnston B. E., Kiani E., Shahinfar S., Sadgrove E. J., Fleming P. J. S. Camera trapping technology and advances: into the new millennium // *Australian Zoologist*. 2020. Vol. 40, No 3. P. 392–403. DOI: 10.7882/AZ.2019.035
- Miao Z., Gaynor K. M., Wang J., Liu Z., Muellerklein O., Norouzzadeh M. S., McInturff A., Bowie R. C. K., Nathan R., Yu S. X., Getz W. M. Insights and approaches using deep learning to classify wildlife // *Scientific Reports*. 2019. Vol. 9, No 1. P. 1–9. DOI: 10.1038/s41598-019-44565-w
- Mohri M., Rostamizadeh A., Talwalkar A. *Foundations of machine learning*. Cambridge, USA: MIT Press, 2012. 505 p.
- Norouzzadeh M. S., Morris D., Beery S., Joshi N., Jovic N., Clune J. A deep active learning system for species identification and counting in camera trap images // *Methods in Ecology and Evolution*. 2021. Vol. 12, No 1. P. 150–161. DOI: 10.1111/2041-210X.13504
- Norouzzadeh M. S., Nguyen A., Kosmala M., Swanson A., Palmer M. S., Packer C., Clune J. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning // *Proceedings of the National Academy of Science*. 2018. Vol. 115, No 25. P. E5716–E5725. DOI: 10.1073/pnas.171936711
- Qin H., Li X., Liang J., Peng Y., Zhang C. DeepFish: Accurate underwater live fish recognition with a

- deep architecture // *Neurocomputing*. 2016. Vol. 187. P. 49–58. DOI: 10.1016/j.neucom.2015.10.122
- Redmon J., Divvala S., Girshick R., Farhadi A. You only look once: unified, real-time object detection // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, USA: IEEE, 2016. P. 779–788. DOI: 10.1109/CVPR.2016.91
- Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks // *arXiv*. 2015. Article: 150601497. DOI: 10.48550/arXiv.1506.01497
- Reyserhove L., Norton B., Desmet P. Best practices for managing and publishing camera trap data. Community review draft. GBIF Secretariat: Copenhagen, 2023. 58 p. DOI: 10.35035/doc-0qzp-2x37
- Rigoudy N., Dussert G., Benyoub A., Besnard A., Birck C., Boyer J., Bollet Y. The DeepFaune initiative: a collaborative effort towards the automatic identification of French fauna in camera-trap images // *bioRxiv*. 2022. DOI: 10.1101/2022.03.15.484324
- Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A. C., Fei-Fei L. Imagenet large scale visual recognition challenge // *International Journal of Computer Vision*. 2015. Vol. 115. P. 211–252. DOI: 10.1007/s11263-015-0816-y
- Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition // *arXiv*. 2014. Article: 1409.1556. DOI: 10.48550/arXiv.1409.1556
- Schneider S., Taylor G. W., Kremer S. Deep learning object detection methods for ecological camera trap data // *15th Conference on Computer and Robot Vision*. Toronto, Canada: IEEE, 2018. P. 321–328. DOI: 10.1109/crv.2018.00052
- Schneider S., Taylor G. W., Kremer S. C. Similarity learning networks for animal individual re-identification – beyond the capabilities of a human observer // *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*. Snowmass, USA: IEEE, 2020. P. 44–52. DOI: 10.1109/WACVW50321.2020.9096925
- Schneider S., Taylor G. W., Linquist S., Kremer S. C. Past, present and future approaches using computer vision for animal re-identification from camera trap data // *Methods in Ecology and Evolution*. 2019. Vol. 10 (4). P. 461–470. DOI: 10.1111/2041-210X.13133
- Shepley A., Falzon G., Meek P., Kwan P. Automated location invariant animal detection in camera trap images using publicly available data sources // *Ecology and Evolution*. 2021. Vol. 11, No 9. P. 4494–4506. DOI: 10.1002/ece3.7344
- Shi C., Xu J., Roberts N.J., Liu D., Jiang G. Individual automatic detection and identification of big cats with the combination of different body parts // *Integrative Zoology*. 2023. Vol. 18 (1). P. 157–168. DOI: 10.1111/1749-4877.12641
- Sener O., Savarese S. Active learning for convolutional neural networks: A core-set approach // *arXiv*. 2018. Article: 1708.00489. DOI: 10.48550/arXiv.1708.00489
- Swanson A., Kosmala M., Lintott C., Simpson R., Smith A., Packer C. Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna // *Scientific Data*. 2015. Vol. 2. P. 1–14. DOI: 10.1038/sdata.2015.26
- Swinnen K. R. R., Reijnders J., Breno M., Leirs H. A novel method to reduce time investment when processing videos from camera trap studies // *PLoS One*. 2014. V. 9. N 6. Article: e98881. DOI: 10.1371/journal.pone.0098881
- Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. Rethinking the inception architecture for computer vision. 2016. URL: [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/papers/Szegedy\\_Rethinking\\_the\\_Inception\\_CVPR\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.pdf) (дата обращения: 22.10.2023).
- Tabak M. A., Norouzzadeh M. S., Wolfson D. W., Newton E. J., Boughto R. K., Ivan J. S., Odell E. A., Newkirk E. S., Conroy R. Y., Stenglein J., Iannarilli F., Erb J., Brook R. K., Davis A. J., Lewis J., Walsh D. P., Beasley J. C., VerCauteren K. C., Clune J., Miller R. S. Improving the accessibility and transferability of machine learning algorithms for identification of animals in camera trap images: MLWIC2 // *Ecology and Evolution*. 2020. Vol. 10, No 19. P. 10374–10383. DOI: 10.1002/ece3.6692
- Tabak M. A., Norouzzadeh M. S., Wolfson D. W., Sweeney S. J., Vercauteren K. C., Snow N. P., Halseth J. M., Di Salvo P. A., Lewis J. S., White M. D., Teton B. Machine learning to classify animal species in camera trap images: applications in ecology // *Methods in Ecology and Evolution*. 2019. Vol. 10, No 4. P. 585–590. DOI: 10.1111/2041-210x.13120
- Tack J. L. P., West B. S., McGowan C. P., Ditchkoff S. S., Reeves S. J., Keever A. C., Grand J. B. AnimalFinder: A semi-automated system for animal detection in time-lapse camera trap images // *Ecological Informatics*. 2016. Vol. 36. P. 145–151. DOI: 10.1016/j.ecoinf.2016.11.003
- Tuia D., Kellenberger B., Beery S., Costelloe B. R., Zuffi S., Risse B., Mathis A., Mathis M. W., Langevelde F.van, Burghardt T., Kays R., Klinck H., Wikelski M., Couzin I. D., Horn G.van, Crofoot M. C., Stewart C. V., Berger-Wolf T. Perspectives in machine learning for wildlife conservation // *Nature Communications*. 2022. Vol. 13, No 1. P. 1–15. DOI: 10.1038/s41467-022-27980-y
- van Gils J. Recognition of wildlife behaviour in camera-trap photographs using machine learning // MSc Thesis *Wildlife Ecology and Conservation*. Netherlands: Wageningen University, 2022. 44 p.
- van Lunteren P. EcoAssist: A no-code platform to train and deploy custom YOLOv5 object detection

- models // *Journal of Open Source Software*. 2023. Vol. 8, No 88. P. 1–3. DOI: 10.21105/joss.05581
- Vélez J., Fieberg J. Guide for using artificial intelligence systems for camera trap data processing. 2022. URL: <https://ai-camtraps.netlify.app> (дата обращения: 26.10.2023).
- Vélez J., McShea W., Shamon H., Castiblanco-Samacho P. J., Tabak M. A., Chalmers C., Fergus P., Fieberg J. An evaluation of platforms for processing camera-trap data using artificial intelligence // *Methods in Ecology and Evolution*. 2023. Vol. 14. P. 459–477. DOI: 10.1111/2041-210X.14044
- Wei W., Luo G., Ran J., Li J. Zilong: A tool to identify empty images in camera-trap data // *Ecological Informatics*. 2020. Vol. 55. P. 1–7. DOI: 10.1016/j.ecoinf.2019.101021
- Whytock R. C., Świeżewski J., Zwerts J. A., Bara-Stupski T., Koumba Pambo A. F., Rogala M., Bahaa-eldin L., Boeke K., Brittain S., Cardoso A. W., Henschel P., Lehmann D., Momboua B., Kiebou Opepa C., Orbell C., Pitman R. T., Robinson H. S., Abernethy K. A. Robust ecological analysis of camera trap data labelled by a machine learning model // *Methods in Ecology and Evolution*. 2021. Vol. 12, No 6. P. 1080–1092. DOI: 10.1111/2041-210X.13576
- Willi M., Pitman R. T., Cardoso A. W., Locke C., Swanson A., Boyer A., Veldthuis M., Fortson L. Identifying animal species in camera trap images using deep learning and citizen science // *Methods in Ecology and Evolution*. 2019. Vol. 10, No 1. P. 80–91. DOI: 10.1111/2041-210X.13099
- Xie Y., Jiang J., Bao H., Zhai P., Zhao Y., Zhou X., Jiang G. Recognition of big mammal species in airborne thermal imaging based on YOLO V5 algorithm // *Integrative Zoology*. 2023. Vol. 18 (2). P. 333–352. DOI: 10.1111/1749-4877.12667
- Xue Y., Wang T., Skidmore A. K. Automatic counting of large mammals from very high resolution panchromatic satellite imagery // *Remote Sensing*. 2017. Vol. 9, No 9. P. 1–16. DOI: 10.3390/rs9090878
- Yosinski J., Clune J., Bengio Y., Lipson H. How transferable are features in deep neural networks? // *arXiv*. 2014. Article: 1411.1792. DOI: 10.48550/arXiv.1411.1792
- Yousif H., Yuan J., Kays R., He Z. Animal Scanner: software for classifying humans, animals, and empty frames in camera trap images // *Ecology and Evolution*. 2019. Vol. 9. P. 1578–1589. DOI: 10.1002/ece3.4747
- Yu X., Wang J., Kays R., Jansen P. A., Wang T., Huang T. Automated identification of animal species in camera trap images // *EURASIP Journal on Image and Video Processing*. 2013. Vol. 52. P. 1–10. DOI: 10.1186/1687-5281-2013-52
- Zhang H., Wu C., Zhang Z., Zhu Y., Zhang Z., Lin H., Sun Y., He T., Mueller J. W., Manmatha R., Li M., Smola A. ResNeSt: Split-Attention Networks // *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. New Orleans, USA: IEEE, 2022. P. 2736–2746. DOI: 10.48550/arXiv.2004.08955

# APPLICATION OF ARTIFICIAL INTELLIGENCE TECHNOLOGIES IN PROCESSING IMAGES FROM CAMERA TRAPS: PRINCIPLES, SOFTWARE, APPROACHES

**OGURTSOV  
Sergey**

*PhD, Central Forest Nature Reserve, A.N. Severtsov Institute of Ecology and Evolution of the Russian Academy of Science (Russia, 172521, Tver region, Nelidovsky city district, Zapovedniy village), etundra@mail.ru*

**EFREMOV  
Vladislav**

*Moscow Institute of Physics and Technology (National Research University) (Russia, 127576, Moscow Region, Moscow), efremov.va@phystech.edu*

**LEUS  
Andrey**

*PhD, Moscow Institute of Physics and Technology (National Research University) (Russia, 141000, Moscow Region, Mytishchi), leus.av@mipt.ru*

**Keywords:** image analysis, detection, classification, computer vision, machine learning, neural networks, pattern recognition, camera traps

**Received on:**  
28 January 2024  
**Published on:**  
02 May 2024

**Summary:** Artificial intelligence (AI) is increasingly penetrating environmental science. This is most rapidly evident in image-based research, for example from drones or camera traps. This review discusses the current development of camera trap research in the application of AI technologies, namely computer vision and deep learning. The basic concepts of Machine Learning (ML) and deep neural networks are briefly considered, which are essential for the modern biologist and ecologist to understand the images processing and analysis. The possibilities of using AI for patterns recognition and object search in images are discussed. An overview of modern software using computer vision and machine learning technologies for recognizing photos and video from camera traps is provided, as well as a brief overview of open data sets for training ML models. In general, eight ML-software are considered: MegaDetector, EcoAssist, MLWIC2, Conservation AI, FasterRCNN+InceptionResNetV2, DeepFaune, ClassifyMe, as well as the first domestic development from the Moscow Institute of Physics and Technology. On the basis of research with camera traps in the Central Forest Nature Reserve, the software was tested and its advantages and disadvantages were identified. In conclusion, the potential of AI application in modern research with camera traps is discussed. This review will be useful for both biologists and ecologists for general acquaintance with neural networks and their application in the field of pattern recognition on images, and for ML-specialists and programmers to understand the applicability of ML models in environmental science and the possibilities of their training on large data sets.