

**ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ В
АГРОПРОМЫШЛЕННОМ КОМПЛЕКСЕ ИРКУТСКОЙ
ОБЛАСТИ ДЛЯ ИДЕНТИФИКАЦИИ СОРНЫХ РАСТЕНИЙ**

Бузина Т.С., кандидат технических наук, доцент,

тел. +79021737301, buzinats@mail.ru

Харламова Е.П., студентка 4 курса,

тел. 89526329961, harlamova.katya.02@mail.ru

ФГБОУ ВО Иркутский ГАУ

***Ключевые слова:** сорные растения, нейронные сети, обучение нейронной сети, градиентный спуск.*

В статье рассматривается использование нейронных сетей и машинного обучения в агропромышленном комплексе для идентификации сорных растений. Описывается подход, основанный на применении сверточных нейронных сетей для обработки изображений сорных растений. Обучение модели выполняется методом градиентного спуска, что позволяет минимизировать ошибку предсказаний. Используется язык программирования Python для разработки нейронной сети. Приводится пример кода, демонстрирующий процесс подготовки изображений, прогнозирования класса изображения и оценку уверенности модели в предсказании.

Одним из важнейших направлений цифровой трансформации агропромышленного комплекса является прикладное использование нейронных сетей и машинного обучения. С помощью использования нейронных сетей можно прогнозировать урожайность сельскохозяйственных культур, а также выявлять сорняки среди полезных культур, изучив множество образцов на основе фотографий. Сорные растения являются серьезной проблемой сельского хозяйства. Они вызывают снижение урожайности культурных растений, лишают

неокрепшие всходы культурных растений влаги и достаточного количества солнечного света [1].

Работа включает в себя анализ, обработку и подготовку данных, выбор и настройку подходящей архитектуры нейронной сети, а также обучение и оценку модели. Полученные результаты позволяют сделать вывод о применимости подхода на основе собранных наблюдений для создания более точных систем искусственного интеллекта. Однако, необходимо провести дальнейшие исследования и эксперименты для улучшения точности и универсальности данного подхода.

Первым этапом разработки нейросетевой модели для определения сорных растений является сбор и анализ информации о видовом разнообразии и распространении сорных растений в регионе. Для исследования выбраны следующие районы Иркутской области: Иркутский, Ангарский, Усольский и Зиминский.

Из всех видов сорных растений Иркутской области:

1) наиболее распространённые сорняки (100%):

- Полевица гигантская;
- Тысячелистник обыкновенный;
- Марь белая;

2) наименее распространённые сорняки (<1%):

1. Щирица запрокинутая;
2. Пикульник ладанниковый, мягковолосый, медунка;
3. Осот шероховатый, острый [2].

За период с июня по сентябрь 2023 года проанализирован 131 вид сорных растений, произрастающих в районах Иркутской области.

Сделаны 2991 фотографии растений, которые размещены в сети iNaturalist в виде наблюдений (рисунок 1). Всего создано 902 наблюдения, каждое из которых содержит около четырех фотографий одного растения.

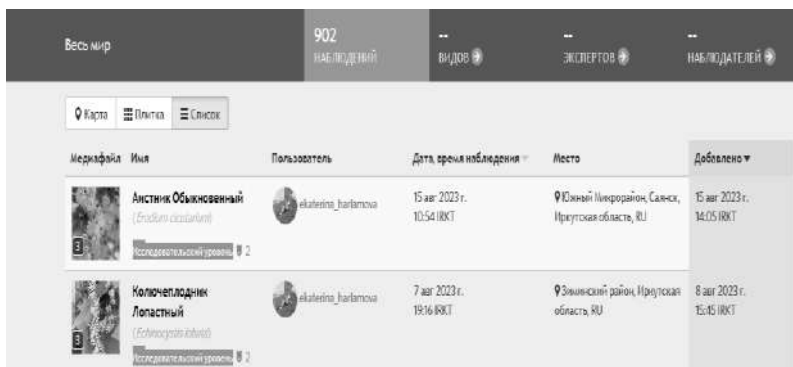


Рис. 1. Фрагмент страницы «Ваши наблюдения» в iNaturalist

Все наблюдения обработаны и систематизированы в таблице, где каждое растение распределено по периоду наблюдения, возрасту, размеру побега и размеру цветка (таблица 1 фрагмент).

Таблица 1

Наблюдения (фрагмент)

Вид растения	Период наблюдения	Возраст растения, дней	Размер побега, см	Размер цветка, см
Бодяк полей	9 июня 2023	50	36	0
	13 июня 2023	55	40	0
Бодяк щетинистый	17 июня 2023	55	57	0
	17 июня 2023	45	45	0
Марь белая	9 июня 2023	12	3	0
	14 июня 2023	12	4	0

Также в наблюдения входят и культурные растения, чтобы не путать их с сорными для большей наглядности предпочтительно разделить их на сорные и не сорные растения. Для этого нужно распределить фотографии и добавить к ним правильное название растения. На рисунке 2 можно увидеть нужное нам распределение [3].

Существует множество архитектур нейросетей, таких как сверточные нейронные сети для обработки изображений и

рекуррентные нейронные сети для работы с последовательными данными. Архитектура нейросети определяет ее структуру и количество слоев, нейронов и соединений между ними. Выбор правильной архитектуры зависит от типа задачи и объема данных.

Для нашей задачи, наиболее подходящей является архитектура сверточной нейронной сети.

Сорняк	Не сорняк
 Астрагал датский	 Венерин Башмачок Вздутый
 Лапчатка Гусиная	 Тимофеевка Луговая
 Иван-чай Узколистный	 Лютышник Щельноносный

Рис. 2. Сорняк/Не сорняк

Сверточная нейронная сеть — класс нейронных сетей, который специализируется на обработке изображений и видео. Такие нейросети хорошо улавливают локальный контекст, когда информация в пространстве непрерывна, то есть её носители находятся рядом [4].

Обучение НС (нейронной сети) осуществляется методом градиентного спуска.

Градиентный спуск — самый используемый алгоритм обучения нейронных сетей, он применяется почти в каждой модели машинного обучения. Метод градиентного спуска с некоторой

модификацией широко используется для обучения персептрона и глубоких нейронных сетей, и известен как метод обратного распространения ошибки.

Суть алгоритма – процесс получения наименьшего значения ошибки. Аналогично это можно рассматривать как спуск во впадину в попытке найти золото на дне ущелья.

Алгоритм обучения сводится к следующей последовательности действий:

1. Подать на вход НС очередной входной вектор из обучающей выборки и определить значения выходов нейронов в выходном слое.

2. Рассчитать $\delta_j^{(n)}$ по формуле (1) и $\Delta w_{ij}^{(n)}$ по формуле (2) для выходного слоя НС

$$\delta_j^{(N)} = \left(y_i^{(N)} - d_i \right) \cdot \frac{dy_j}{ds_j}, \quad (1)$$

где $\delta_j^{(n)}$ - величина слоя n ; y_i - значение выхода j -го нейрона; d_i - желаемое значение j -го выхода; $\frac{dy_j}{ds_j}$ - значение производной функции активации по ее аргументу для нейрона j .

$$\Delta w_{ij}^{(n)} = -n \cdot \delta_j^{(n)} \cdot x_i^n, \quad (2)$$

где W_{ij} - весовой коэффициент синаптической связи, соединяющей i -ый нейрон слоя $n-1$ с j -ым нейроном слоя n ; n - коэффициент скорости обучения, $0 < n < 1$; x_i - значение i -го выхода нейрона.

3. Рассчитать $\delta_j^{(n)}$ и $\Delta w_{ij}^{(n)}$ по формулам (3) и (2) для остальных слоёв НС

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} \cdot w_{jk}^{(n+1)} \right] \cdot \frac{dy_j}{ds_j}, \quad (3)$$

где $\delta_j^{(n)}$ - величина слоя n из величин $\delta_k^{(n+1)}$ более старшего слоя $n+1$; k - число нейронов в слое $n+1$.

4. Скорректировать все веса НС:

$$\Delta w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t), \quad (4)$$

где t – номер текущей итерации.

5. Если ошибка существенна, то перейти на шаг 1.

На этапе 2 сети поочередно в случайном порядке предъявляются вектора из обучающей последовательности [5].

В качестве языка программирования для работы выбран Python, как самый известный язык программирования. Причин у этого две: простота и гибкость.

Популярность Python обусловлена обширной коллекцией доступных библиотек и фреймворков. Такие библиотеки, как TensorFlow, PyTorch и Keras, позволяют создавать сложные модели по типу ChatGPT и LLaMA.

Начнем с импорта нужных библиотек, необходимых для разработки нашей нейронной сети.

Нам понадобятся:

- Keras- библиотека глубокого обучения на Python, представляющая собой высокоуровневый API;
- Python Imaging Library- библиотека языка Python, предназначенная для работы с растровой графикой;
- NumPy - библиотека Python, которая предоставляет объект многомерного массива, различные производные объекты и набор процедур для быстрых операций с массивами;
- TensorFlow - открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия.

```
from keras.models import load_model
from PIL import Image, ImageOps
```

```
import numpy as np
```

```
import tensorflow
```

Далее загружаем модель и метки классов.

```
model = load_model("keras_model.h5", compile=False)
```

```
class_names = open("labels.txt", "r").readlines()
```

Подготавливаем изображения. Изображение загружается, изменяется его размер и нормализуется для подачи на выход модели.

```
image = Image.open("/content/ф26.jpg").convert("RGB")
```

```
size = (224, 224)
```

```
image = ImageOps.fit(image, size, Image.Resampling.LANCZOS)
```

```
image_array = np.asarray(image)
```

```
normalized_image_array = (image_array.astype(np.float32) /  
127.5) - 1
```

```
data[0] = normalized_image_array
```

Происходит прогнозирование класса. Модель используется для предсказания класса изображения.

```
prediction = model.predict(data)
```

```
index = np.argmax(prediction)
```

```
class_name = class_names[index]
```

```
confidence_score = prediction[0][index]
```

Выводим результат. Выводится предсказанный класс и уверенность в предсказании.

```
print("Class:", class_name[2:], end="")
```

```
print("Confidence Score:", confidence_score)
```

На рисунке 5 представлен код, выполняющий предсказание класса изображения, который выводит уверенность в этом предсказании.

Из рисунка 5 видно, как определился класс Бодяк щетинистый и его показатель уверенности 0.9999087.

Таким образом, значение 0.9999087 означает, что модель считает предсказание почти абсолютно верным, с вероятностью около 99,9%.

Применение нейросетевых технологий в изучении сорных растений позволяет автоматизировать процессы идентификации и классификации видов, улучшить точность и скорость обработки данных, создавать прогнозы распространения сорных растений и моделировать их динамику.

```
from keras.models import load_model
from PIL import Image, ImageOps
import numpy as np
import tensorflow
model=tensorflow.keras.models.load_model('/content/keras_model.h5')
np.set_printoptions(suppress=True)

model = load_model("keras_model.h5", compile=False)

class_names = open("labels.txt", "r").readlines()

data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

image = Image.open("/content/026.jpg").convert("RGB")
size = (224, 224)
image = ImageOps.fit(image, size, Image.Resampling.LANCZOS)

image_array = np.asarray(image)

normalized_image_array = (image_array.astype(np.float32) / 127.5) - 1

data[0] = normalized_image_array

prediction = model.predict(data)
index = np.argmax(prediction)
class_name = class_names[index]
confidence_score = prediction[0][index]

print("Class:", class_name[2:], end="")
print("Confidence Score:", confidence_score)

WARNING:tensorflow:No training configuration found in the save file, so the model was *not* compiled. Compile it manually.
1/3 [=====] - 2s 2s/step
Class: 56дкн щитовидный
Confidence Score: 0.9999087
```

Рис. 5. Код нейронной сети

Это способствует развитию эффективных методов контроля над сорняками, оптимизации сельскохозяйственного производства и сохранению биоразнообразия.

Библиографический список

1. Классификация сорняков сельскохозяйственных культур и меры борьбы с ними: [Электронный ресурс]. - Режим доступа <https://www.cropscience.bayer.ru/sorniaki-siel-skokhoziaistviennykh-kul-tur>.
2. Сорняки Иркутской области: [Электронный ресурс]. - Режим доступа: <https://old.donorost.ru/maps/country/rossiya/region/irkutskayaoblast/type/weed>.
3. Харламова, Е. П. Применение нейросетевых технологий для изучения сорных растений / Е. П. Харламова // Значение научных

студенческих кружков в инновационном развитии агропромышленного комплекса региона : Сборник научных тезисов студентов, п. Молодежный, 12–13 октября 2023 года. – п. Молодежный: Иркутский государственный аграрный университет им. А.А. Ежевского, 2023. – С. 507-508.

4. Сверточная нейронная сеть: как устроена, архитектуры и параметры-использование сверточных нейронных сетей: [Электронный ресурс]. – Режим доступа: <https://practicum.yandex.ru/blog/svertochnye-neyronnye-seti/>.

5. Алгоритм обратного распространения ошибки: [Электронный ресурс]. – Режим доступа: https://studopedia.su/10_114362_algoritm-obratnogo-rasprostraneniya-oshibki.html.

THE USE OF NEURAL NETWORKS IN THE AGRO-INDUSTRIAL COMPLEX OF THE IRKUTSK REGION FOR THE IDENTIFICATION OF WEEDS

Buzina T. S., Kharlamova E.P.

Keywords: *weeds, neural networks, neural network training, gradient descent.*

The article discusses the use of neural networks and machine learning in the agro-industrial complex for the identification of weeds. An approach based on the use of convolutional neural networks for image processing of weeds is described. The model is trained using the gradient descent method, which minimizes the prediction error. The Python programming language is used to develop a neural network. An example code is provided that demonstrates the process of preparing images, predicting the image class, and evaluating the confidence of the model in prediction.