

УДК 004.032.26; 631.574

**АУГМЕНТАЦИЯ БАЗЫ ИЗОБРАЖЕНИЙ ЛИСТЬЕВ ЯБЛОНИ ДЛЯ
ДОСТИЖЕНИЯ МАКСИМАЛЬНОЙ ТОЧНОСТИ ОПРЕДЕЛЕНИЯ
КЛАССА ЗАБОЛЕВАНИЯ ПРИ ОБУЧЕНИИ НЕЙРОННОЙ СЕТИ (НС)**

Анатолий Иванович Бутенко

доктор сельскохозяйственных наук, профессор

but_tolik@mail.ru

Максим Геннадиевич Тимофеев

студент

vielseitig@mail.ru

Мичуринский государственный аграрный университет

г. Мичуринск, Россия

Аннотация. В данной статье представлена аугментация, как методика наращивания базы данных.

Ключевые слова: аугментация, tensorflow, генератор изображений, keras

В современном мире для организации систем автоматизированного производства под определенные задачи в сельском хозяйстве активно используются информационные технологии на базе машинного обучения [1-3].

Целью обучения машины является достижением максимальной точности выполнения сложного, трудоемкого (рутинного) процесса без участия человека, а сам процесс обучения можно разложить на несколько этапов:

1. Определение цели автоматизации процесса.
2. Сбор базы данных.
3. Создание и обучение прототипа НС.

Точность работы НС зависит не только от ее архитектуры, но и от качества и размера собранной базы данных и если данных недостаточно, то применяют методику аугментации для их увеличения [4, 5].

Аугментация (англ. data augmentation) – это метод наращивания дополнительных данных в процессе изменения имеющихся (базовых) данных.

Инструментом для машинного обучения применяется библиотека с открытым исходным кодом - TensorFlow, созданная для высокоуровневого, объектно-ориентированного языка программирования – Python.

Библиотека TensorFlow компилирует множество различных алгоритмов и моделей, что позволяет реализовывать глубокие НС для задач по распознаванию и классификации изображений.

Для реализации метода аугментации применяется генератор изображений ImageDataGenerator() в модуле TensorFlow.keras.preprocessing.image (рисунок 1).

```
1 tf.keras.preprocessing.image.ImageDataGenerator(featurewise_center=False, samplewise_center=False,  
2     featurewise_std_normalization=False, samplewise_std_normalization=False, zca_whitening=False,  
3     zca_epsilon=1e-06, rotation_range=0, width_shift_range=0.0, height_shift_range=0.0,  
4     brightness_range=None, shear_range=0.0, zoom_range=0.0, channel_shift_range=0.0,  
5     fill_mode='nearest', cval=0.0, horizontal_flip=False, vertical_flip=False, rescale=None,  
6     preprocessing_function=None, data_format=None, validation_split=0.0, dtype=None)
```

Рисунок 1 – Генератор изображений

На вход функции генератора подаются параметры, по которым будет производиться изменение базового изображения и создаваться похожее изображение [6].

Алгоритм генерации десяти изображений для одного файла можно разложить на 4 этапа (рисунок 2):

1. Загрузка (англ. import) библиотек и модулей.
2. Создание генератора с активацией необходимых параметров.
3. Обработка изображения для подачи в генератор.
4. Генерация изображений.

```
1 # 1. import библиотек и модулей:
2 import numpy as np           # NumPy - библиотека для работы с многомерными массивами массивами
3 import keras                 # keras - библиотека для разработки искусственных нейронных сетей
4 # ImageDataGenerator - генератор изображений
5 # load_img - функция загрузки изображения с google-диска
6 # img_to_array - функция для преобразования загруженного изображения в формате PIL в массив NumPy
7 # array_to_img - функция для преобразования NumPy-массива в изображение формата PIL
8 from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
9 # 2. Создаем генератор для генерации изображений
10 datagen = ImageDataGenerator(rotation_range=45, # Поворачиваем изображения при генерации выборки
11                               width_shift_range=0.2, # Двигаем изображения по ширине при генерации выборки
12                               height_shift_range=0.2, # Двигаем изображения по высоте при генерации выборки
13                               rescale=1./255, # Значения цвета меняем на дробные показания
14                               shear_range=0.2, # Диапазон сдвига пикселей изображения
15                               zoom_range=0.2, # Зумируем изображения при генерации выборки
16                               horizontal_flip = True, # Отзеркаливание изображений по горизонтали
17                               vertical_flip=True, # Отзеркаливание изображений по вертикали
18                               fill_mode = 'nearest', # Заполнение пикселей вне границ ввода
19                               brightness_range=[0.5, 1.5]) # Диапазон выбора сдвига по яркости
20 # 3. Обработка изображения:
21 # С помощью функции load_img копируем очередное jpg-файл и сохраняем в переменную img
22 img = load_img(r"/content/drive/MyDrive/База/sheet/sheet (1).JPG")
23 # С помощью функции img_to_array преобразуем jpg-файл в NumPy-массив и запишем в переменную X
24 x = img_to_array(img)
25 # С помощью функции reshape() изменим размерность NumPy-массива в переменной X
26 x = x.reshape((1,) + x.shape)
27 # 4. Генерация изображений:
28 i = 0 # Счетчик количества batch
29 # В цикле перебора каждый параметр генератора изображений применим к NumPy-массиву, преобразуем в
30 # jpg-файл и сохраним в каталог с изображениями на google-диске
31 for batch in datagen.flow(x, batch_size=1, save_to_dir =r'/content/drive/MyDrive/База/sheet',
32                            save_prefix = 'sheet', save_format='jpg'):
33     i+=1 # Считаем изображения
34     if i>4: # Если i более 4
35         break # стоп цикл
```

Рисунок 2 – Программа для генерации изображений

Результат сгенерированных изображений с различными значениями параметров генератора (рисунок 3).



Рисунок 3 – Сгенерированные изображения (1-е слева - исходное изображение).

Список литературы:

1. Коротков А.А., Астапов А.Ю., Криволапов И.П. Элементы технологий точного земледелия // Наука и Образование. 2020. Т. 3. № 3. С. 132.
2. Астапов А.Ю., Криволапов И.П., Акишин Д.В. Оптический метод определения степени зрелости плодов яблони // Наука в центральной России. 2019. № 6 (42). С. 17-22.
3. Bonitet assessment of graft-rootstock combinations of apple tree varieties in the conditions of the central Chernozem region / Z.N. Tarova, L.V. Bobrovich, I.P. Krivolapov, A.A. Korotkov, O.A. Borisova // В сборнике: IOP Conference Series: Earth and Environmental Science. Сер. "International Conference on Agricultural Science and Engineering" 2021. С. 012012.
4. Русскоязычная документация Keras [Электронный ресурс] Режим доступа: <https://ru-keras.com/>, свободный. – (дата обращения: 22.03.2022).
5. TensorFlow Core v2.8.0. ImageDataGenerator – техническая информация [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator, свободный. – (дата обращения: 22.03.2022).
6. Элбон Крис Машинное обучение с использованием Python. Сборник рецептов: Пер. с англ. СПб.: БХВ-Петербург, 2019. 384 с.

UDC 004.032.26; 631.574

AUGMENTATION OF THE IMAGE BASE OF APPLE LEAVES TO ACHIEVE MAXIMUM ACCURACY IN DETERMINING THE CLASS OF THE DISEASE WHEN TRAINING A NEURAL NETWORK (NN)

Anatoly I. Butenko

Doctor of Agricultural Sciences, Professor

but_tolik@mail.ru

Maxim G. Timofeev

student

vielseitig@mail.ru

Michurinsk State Agrarian University

Michurinsk, Russia

Annotation. This article presents augmentation as a technique for increasing the database.

Key words: augmentation, tensorflow, image generator, keras

Статья поступила в редакцию 07.05.2022; одобрена после рецензирования 09.06.2022; принята к публикации 30.06.2022.

The article was submitted 07.05.2022; approved after reviewing 09.06.2022; accepted for publication 30.06.2022.