

УДК 004.032.26:58

DOI 10.34822/1999-7604-2021-1-41-50

ОЦЕНКА СОСТОЯНИЯ РАСТЕНИЙ С ИСПОЛЬЗОВАНИЕМ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

М. Я. Брагинский ✉, **Д. В. Тараканов**

Сургутский государственный университет, Сургут, Россия

✉ E-mail: mick17@mail.ru

Данная работа продолжает исследования авторов в области построения систем автоматической диагностики состояний и динамики роста растений. Построение подобного рода систем требует решения ряда задач, связанных с обработкой цифровых изображений, таких как детектирование и идентификация состояния растений. В качестве центральной компоненты рассматриваемой системы используется математический аппарат сверточных нейронных сетей с описанием архитектуры, процедуры обучения и тестирования нейронной сети, а также результатов эксперимента. Анализ работы сверточных нейронных сетей показывает высокую эффективность предлагаемого решения поставленной задачи оценки состояния биологических культур.

Ключевые слова: сверточные нейронные сети, сегментация изображения, классификация.

ESTIMATION OF PLANTS HEALTH USING CONVOLUTIONAL NEURAL NETWORKS

M. Ya. Braginsky ✉, **D. V. Tarakanov**

Surgut State University, Surgut, Russia

✉ E-mail: mick17@mail.ru

The article continues the authors' research in the field of building systems for automatic diagnostics conditions and dynamics of plant growth. New concepts of intelligent farming, where field conditions are controlled by autonomous systems, have now become widely used. Building such systems requires solving several problems related to digital image processing, such as detecting plants and identifying their condition. Therefore, the article studies the actual problem of constructing an automatic system for assessing the state and dynamics of plant growth. As the central component of the system under consideration, the authors propose to use the mathematical apparatus of convolutional neural networks. The article describes the architecture, training procedure, and testing of the neural network. The results of computer experiments are presented. The analysis of the work of convolutional neural networks shows the high efficiency of the proposed solution to the problem of assessing the state and growth dynamics of biological crops.

Keywords: convolutional neural networks, image segmentation, classification.

Введение

Контроль полевых условий с помощью автономных систем как одно из направлений цифровизации широко применяется в современных агрокомплексах, поскольку идентификация и своевременное лечение заболеваний растений, которые в большинстве случаев проявляются на листьях и стеблях, могут предотвратить финансовые потери. Велика потребность в новых технологиях, которые отслеживают и прогнозируют воздействие абиотического (факторы неживой среды) и биотического (влияние живых существ) стресса на рост и продуктивность растений, поэтому идентификация листьев и стеблей, оценка доли поражения растений играют ключевую роль в успешном выращивании сельскохозяйственных культур [1–2]. Для обработки изображений, выявления и фиксации участков поражения растений целесообразно использовать метод автоматического обнаружения [3–5].

Изображение может быть получено с помощью обычной цифровой камеры в режиме реального времени. Центральной задачей системы автоматического детектирования заболеваний и роста растений является выделение информационных признаков и сегментация изображений. В нашем случае для решения поставленной задачи используется пакет прикладных программ MatLab.

Архитектура системы оценки состояния растений

На рис. 1 представлена схема оценки состояния растения (здоровое/больное) с идентификацией 12 типовых заболеваний, динамики роста и биомассы растительных культур.

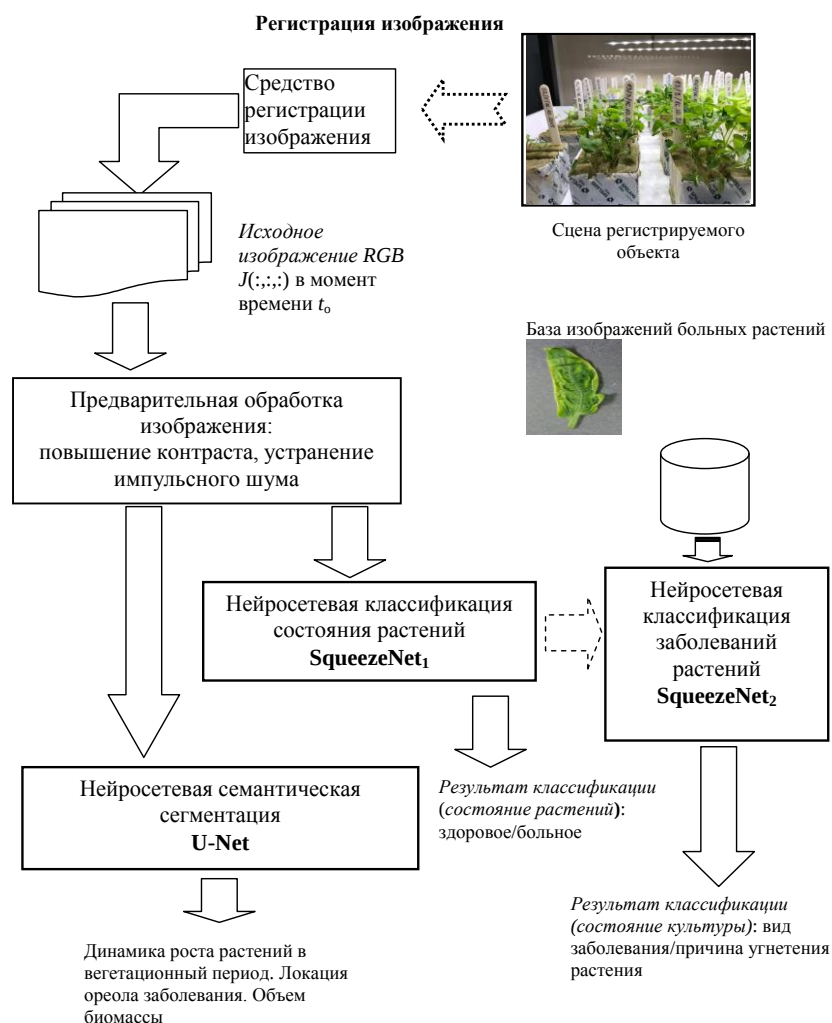


Рис. 1. Этапы оценки состояния растения с использованием сверточных нейронных сетей
Примечание: составлено авторами.

Для решения поставленной задачи была выбрана сверточная нейронная сеть SqueezeNet, обладающая высокой скоростью обучения и не требующая для работы большого объема оперативной памяти.

Архитектура сети имеет следующий вид [6–8]: сеть делится на 8 одинаковых блоков. В блоке на рис. 2 параллельно выполняются свертки 1×1 и 3×3 . На начальном этапе выполняется свертка 1×1 с целью сжатия изображения. Этот слой свертки называется squeeze-слоем (слоем сжатия).

Также имеется слой субдискретизации average pooling, что позволяет уменьшить количество компонентов сети.

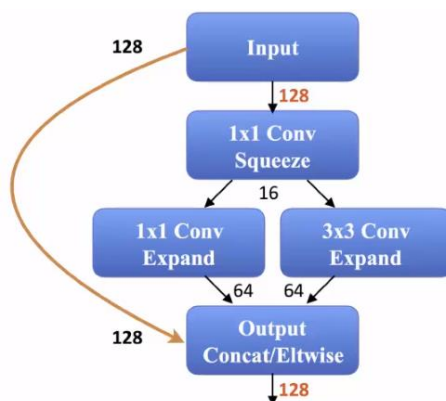


Рис. 2. Структура блока сети SqueezeNet [8]

Для загрузки сети SqueezeNet (рис. 3) в среде MATLAB необходимо выполнить команду:

```
>> net = squeezeNet

net =

DAGNetwork with properties:

    Layers: [68x1 nnet.cnn.layer.Layer]
Connections: [75x2 table]
InputNames: {'data'}
OutputNames: {'ClassificationLayer_predictions'}
```

Рис. 3. Загрузка предобученной сети в среде MATLAB

Примечание: составлено авторами.

После загрузки сети необходимо выполнить процедуру замены последних слоев. На рис. 4 выделены слои, подлежащие корректировке.

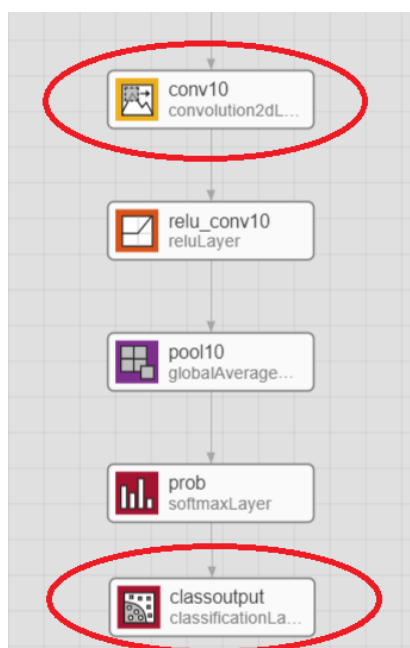


Рис. 4. Слои, подлежащие изменению

Примечание: составлено авторами.

В случае SqueezeNet замене подлежит слой свертки, поскольку изначально сеть обучена классификации на 1 000 классов и этот слой свертки содержит 1 000 сверток размера 1×1 . Так как целью работы является классификация растения на 2 класса (больное/здоровое), необходимо этот слой свертки заменить аналогичным, но с другими параметрами, а также заменить классификационный слой (рис. 5).

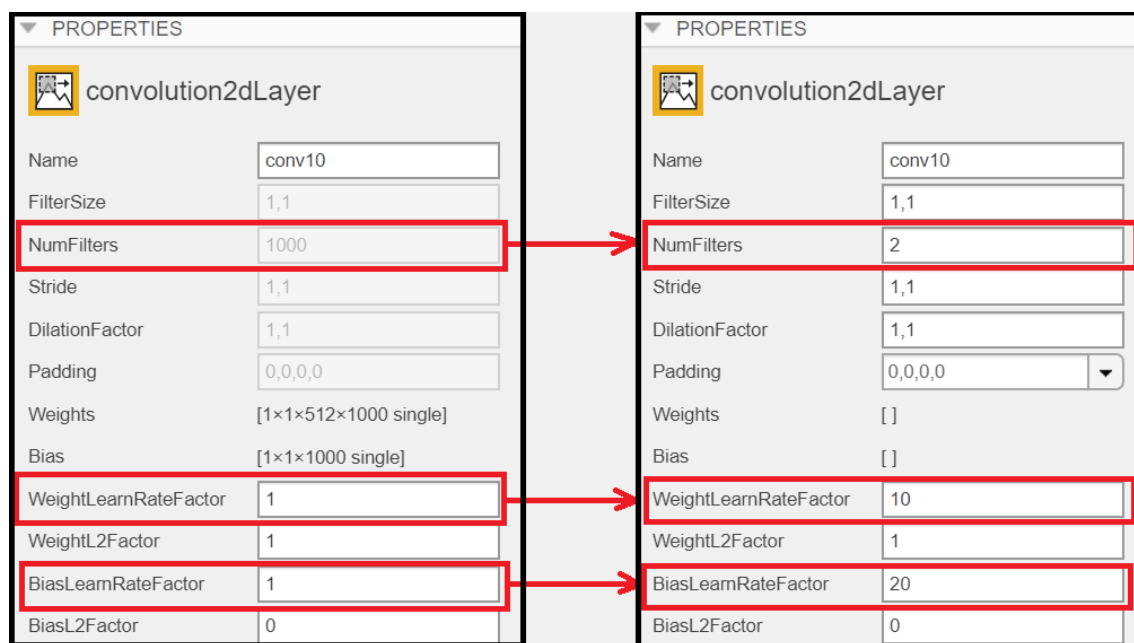


Рис. 5. Замена параметров слоя свертки
Примечание: составлено авторами.

На рис. 5 показаны изменения параметра NumFilters (количество фильтров свертки, с 1 000 на 2), а также параметров мультипликатора WeightLearnRateFactor (с 1 на 10) и BiasLearnRateFactor (с 1 на 20). Мультипликатор показывает отношение скорости обучения отдельного слоя к скорости обучения всей сети.

Следует также изменить параметры второго слоя – слоя классификации, – поскольку в исходном варианте он проводит классификацию на 1 000 классов, а в данной работе необходимо провести ее на 2 класса (рис. 6).

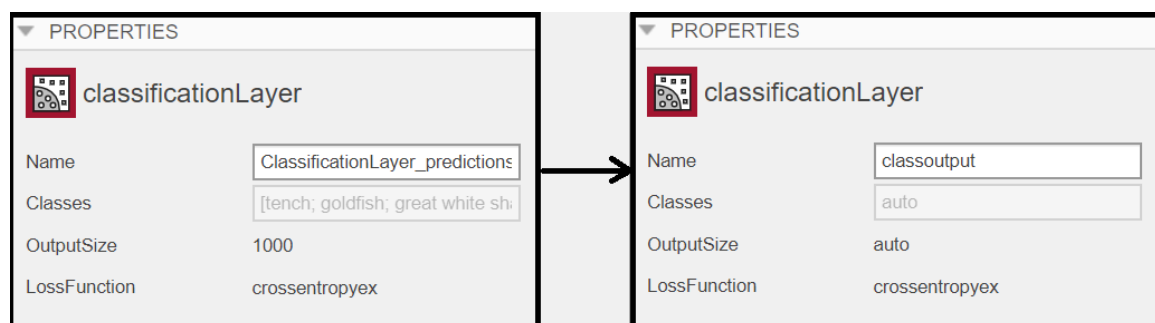


Рис. 6. Замена классификационного слоя
Примечание: составлено авторами.

Следует отметить, что при смене классификационного слоя требуемые параметры выставляются автоматически. После замены необходимых слоев можно экспортировать скорректированную сеть в рабочее пространство MATLAB.

После выбора сети и замены последних обучаемых слоев сеть передается на обучение.

Тип обучения – управляемое (с учителем). Для этого была подготовлена обучающая выборка из изображений растений, разделенных на 2 класса: растения, пораженные болезнями, и здоровые растения.

Код программы для обучения скорректированной сети в среде MATLAB представлен на рис. 7.

```
path = 'ImageFolder';
imds =
imageDatastore(path,'IncludeSubfolders',1,'LabelSource','foldernames');
tbl = countEachLabel(imds)
%%
imds = splitEachLabel(imds,1360);
imds.ReadFcn = @readFunction;
countEachLabel(imds)
[trainingDS,testDS] = splitEachLabel(imds,1100,260,'randomize');
trainingDS.Labels = categorical(trainingDS.Labels);
testDS.Labels = categorical(testDS.Labels);
%%
miniBatchSize = 20;%
numImages = numel(trainingDS.Files);
maxEpochs = 18;%
lr = 0.0005;
opts = trainingOptions('sgdm',...
    'InitialLearnRate',lr,...
    'MaxEpochs',maxEpochs,...
    'MiniBatchSize',miniBatchSize,...
    'LearnRateSchedule','piecewise',...
    'LearnRateDropPeriod',6,...
    'LearnRateDropFactor',0.2,...
    'Plots','training-progress');
net = trainNetwork(trainingDS,lgraph_1,opts);
%%
tic
[labels,err_test] = classify(net,testDS,'MiniBatchSize',20);
toc
```

Рис. 7. Код программы для обучения скорректированной сети

Примечание: составлено авторами.

Вначале с помощью функции «ImageDatastore» необходимо указать, в какой директории хранятся изображения из обучающей выборки. В качестве параметров этой функции выступает переменная «path», хранящая путь к изображениям:

```
path = 'ImageFolder';
imds = imageDatastore(path,'IncludeSubfolders',1,'LabelSource','foldernames');
```

Параметр «IncludeSubfolders» означает, что в переданном пути находятся папки с изображениями. Последние два параметра «LabelSource» и «foldernames» означают, что имена классов изображений будут совпадать с именами папок. Результат выполнения функции записывается в переменную «imds».

На рис. 8, 9 показаны примеры обучающих изображений каждого класса.



Рис. 8. Пример обучающего изображения листа больного растения (класс «disease»)

Примечание: фото авторов.



Рис. 9. Пример обучающего изображения листа здорового растения (класс «healthy»)

Примечание: фото авторов.

Так как на вход сети должны подаваться изображения размером $[227 \times 227]$, то для необходимого преобразования входных изображений в среде MATLAB используется функция «imresize». На следующем шаге обучающая выборка изображений функцией «splitEachLabel» делится на тестовую и тренировочную выборки:

```
countEachLabel(imds)
[trainingDS,testDS] = splitEachLabel(imds,1100,260,'randomize');
```

В качестве аргументов функции используется имя обучающей выборки, а также количество изображений каждого класса, входящих в тестовую и тренировочную выборки. Параметр «randomize» означает, что изображения распределяются случайным образом.

Для обучения сети в переменную «opts» записывается массив «trainingOptions», в котором в качестве обучающего алгоритма выбирается алгоритм «sgdm» (стохастический градиентный спуск), минимальный размер батча устанавливается равным 20 изображениям; максимальное количество эпох – 18; скорость обучения (один из параметров градиентного спуска) – 0.005.

Непосредственно обучение выполняется функцией «trainNetwork», в которую передаются массив заданных параметров «opts», тренировочная выборка «trainingDS» и сеть «lgraph_1» с замененными слоями свертки и классификации:

```
net = trainNetwork(trainingDS,lgraph_1,opts);
```

После этого происходит процедура обучения сети «net» (рис. 10).

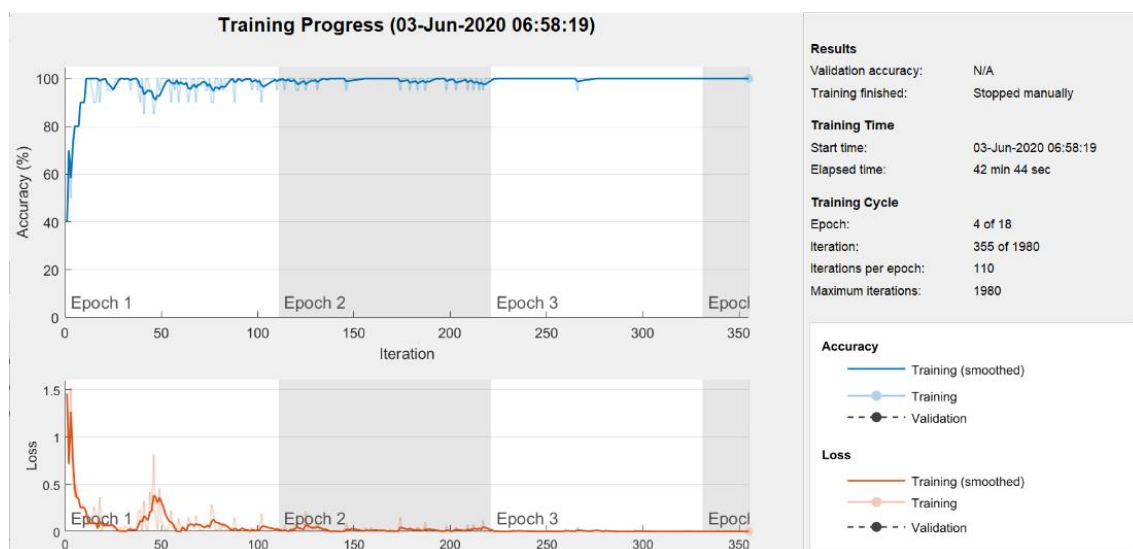


Рис. 10. Результат обучения сверточной нейронной сети

Примечание: составлено авторами.

Как видно из рис. 10, обучение было остановлено уже на четвертой эпохе, поскольку сеть за это время достигла максимальной точности.

Теперь необходимо оценить точность сети на тестовом множестве:

```
tic  
[labels,err_test] = classify(net,testDS,'MiniBatchSize',20);  
toc
```

Как видно по этому коду, оценка точности происходит с помощью функции «classify», принимающей в качестве аргументов обученную сеть, минимальный размер батча и тестовую выборку «testDS», содержащую 520 изображений (по 260 из каждого класса). Время оценки замеряется посредством встроенных в MATLAB функций «tic» и «toc» (рис. 11).

```
Elapsed time is 42.855568 seconds.  
  
ans =  
  
0.9147
```

Рис. 11. Результат оценки точности сети на тестовой выборке

Примечание: составлено авторами.

Точность была оценена примерно за 43 секунды и составила 91,47 %. При необходимости этот показатель можно повысить путем оптимизации параметров обучения. В случае со SqueezeNet оптимальные параметры повышают точность до 94 %.

Тестирование обученной сети

На первом этапе будет произведена классификация пораженного растения (рис. 12). Данное изображение было получено на локальной ферме Сургутского государственного университета (СурГУ).



Рис. 12. Классифицируемое изображение с пораженным растением

Примечание: фото авторов.

Вначале производится считывание изображения при помощи встроенной в MATLAB функции «imread» и его запись в переменную I:

```
>> I = imread("image1.jpeg");
```

Затем считанное изображение встроенной функцией «imresize» преобразуется в требуемый формат для обработки сетью (для SqueezeNet размер составляет 227×227 пикселей):

```
>> I = imresize(I,[227 227]);
```

Функцией «classify», принимающей в качестве аргументов имена сети и изображения, производится классификация, при этом в переменную «labels» записывается имя класса, а в «err_test» – вероятности принадлежности изображения соответствующим классам:

```
>> [labels,err_test] = classify(net,I);
```

Встроенная функция «imshow» выводит изображение на экран. Для отображения результата выводится конкатенация заголовка, которому присвоено имя класса, и вероятности принадлежности изображения данному классу:

```
>> imshow(I)
>> label = labels;
>> title(string(label) + ", " + num2str(100*max(probs),3) + "%");
```

На рис. 13 продемонстрирован результат классификации состояния растения по исходному изображению.

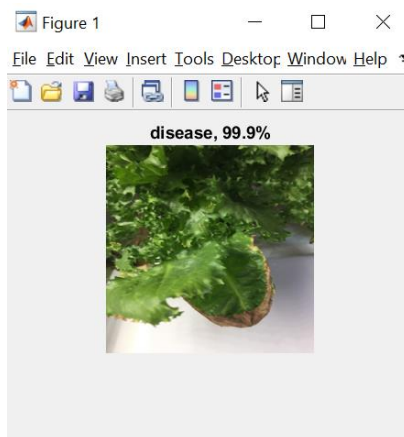


Рис. 13. Классифицированное изображение больного растения

Примечание: скриншот авторов.

Аналогичным образом была проведена оценка состояния здоровых растений, изображения которых также были получены на локальной ферме СурГУ (рис. 14, 15).



Рис. 14. Классифицируемое изображение со здоровым растением № 1

Примечание: фото авторов.



Рис. 15. Классифицируемое изображение со здоровым растением № 2

Примечание: фото авторов.

На рис. 16, 17 продемонстрированы результаты классификации здорового растения.



Рис. 16. Классифицированное изображение здорового растения № 1

Примечание: скриншот авторов.

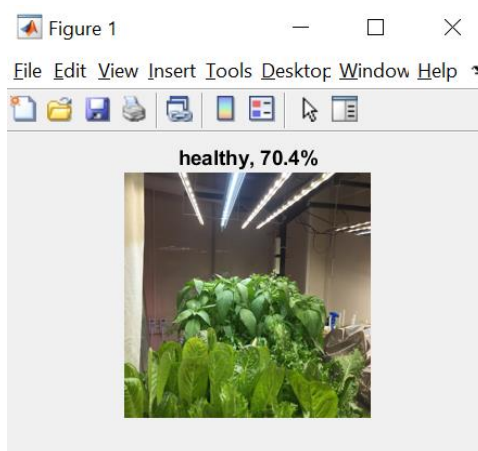


Рис. 17. Классифицированное изображение здорового растения № 2

Примечание: скриншот авторов.

Как видно из рис. 13, 16 и 17, сверточная нейронная сеть успешно решила задачу оценки состояния растений.

Заключение

В данной работе описано решение задачи классификации состояния растений средствами машинного обучения. Исследования выполнялись с использованием базы цветных фотографий. Для решения поставленных задач было использовано два типа сверточных нейронных сетей. Были рассмотрены и протестированы технологии передачи обучения и семантической сегментации изображения, в результате чего получена нейронная сеть, решающая задачу классификации растений. Для обучения сети семантической сегментации был рассмотрен метод получения сегментированных масок путем пороговой бинаризации цветного изображения. Для детектирования состояния растений был подготовлен датасет из 3 000 изображений растений, разделенных на 2 класса – больные и здоровые. Результаты представленной работы могут быть использованы в современных автоматических агрокомплексах.

Литература

1. Ashok R., Uma S. K. Garden Environmental Monitoring & Automatic Control System Using Sensors // International Journal for Modern Trends in Science and Technology (IJMTST). 2016. Vol. 2, No. 5. P. 141–144.
2. Faouzi D., Bibi-Triki N., Draoui B., Abène A. Greenhouse Environmental Control Using Optimized, Modeled and Simulated Fuzzy Logic Controller Technique in MATLAB SIMULINK // Computer Technology and Application. 2016. No. 7. P. 273–286.
3. Albadarneh A., Ahmad A. Automated Flower Species Detection and Recognition from Digital Images // IJCSNS International Journal of Computer Science and Network Security. 2017. Vol. 17, No. 4. P. 144–151.
4. Elangovan K., Nalini S. Plant Disease Classification Using Image Segmentation and SVM Techniques // International Journal of Computational Intelligence Research. 2017. Vol. 13, No. 7. P. 1821–1828.
5. Hagara M., Kubinec P. About Edge Detection in Digital Images // Radioengineering. 2018. Vol. 27, No. 4. P. 919–929.
6. Setiawan W., Utoyo M. I., Rulaningtyas R. Reconfiguration Layers of Convolutional Neural Network for Fundus Patches Classification // Bulletin of Electrical Engineering and Informatics. 2021. No. 10 (1). P. 383–389.
7. Kaggle Datasets. URL: <https://www.kaggle.com/tags/plants> (дата обращения: 19.03.2021).
8. SqueezeNet-Residual by songhan. URL: <http://songhan.github.io/SqueezeNet-Residual> (дата обращения: 19.03.2021).