BULGARIAN ACADEMY OF SCIENCES
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS • 2020 • Vol. **74**, pp. 41-50
p-ISSN: 2738-7356; e-ISSN: ISSN: 2738-7364
https://doi.org/10.7546/PECR.74.20.05

DCNN against Wheat Weeds

Asya Toskova[1], *Zlatina Uhr*[2], *Stanimir Stoyanov*[1], *Borislav Toskov*[1]

[1] *Plovdiv University "Paisii Hilendarski", Plovdiv, Bulgaria*
[2] *Institute of Plant Genetic Resources "K. Malkov", Sadovo, Bulgaria*
Emails: asya_toskova@uni-plovdiv.bg, zlatinapg@abv.bg, stani@uni-plovdiv.net, toskov@uni-plovdiv.net

Abstract: The article presents a classification model for detecting some common and economically important weeds in wheat crops in Bulgaria. For this purpose a deep convolutional neural network (DCNN) has been created. The dataset contains full-color images of wheat and six early-stage weeds. The data represents a selected and processed part of the existing V2 Plant Seedlings Dataset. DCNN is trained to recognize 7 classes. The network is trained on images from scratch. The model successfully classifies the test set with an accuracy of 92%. It is expected that applying the classifier in a real work environment will automate and speed up the process of identifying weeds and will be useful for both their early removal and limiting the amount of harmful herbicides.

Keywords: *Deep Convolutional Neural Network, Weed detection, Common wheat, Intelligent agriculture.*

1. Introduction

One of the most frequently grown crops worldwide is the wheat. Each year about 15 million acres of wheat are cultivated in Bulgaria. The good yield and the healthy harvest are natural goals for producers. Although the wheat is an enduring crop, its cultivation depends on a number of environmental factors. Weeds appear to be harmful ones. They are unpretentious, they multiply rapidly, successfully compete with the crop for water, nutrients and light, and poison the wheat with the allelochemicals that they exude. This is detrimental to its development, especially in the first few weeks of its life cycle. In addition to hindering the growth of wheat, weeds are a prerequisite for the development of diseases and pests. Therefore, it is particularly important that harmful plants are discovered and removed by the end of the "tillering" phase (wheat growing phase) [1].

The effective struggle with weed associations requires the collection and processing of certain data, the deciding on an appropriate action and the precise implementation of the chosen action. A classic practice for weed detection is to traverse the planted field and to map the species composition and the density of each species [2]. After processing the collected information, weed species are extirpated with suitable herbicides. In order to protect the crop plants and the environment, the spraying with harmful chemicals is carried out in a differentiated manner.

The new high-tech opportunities allow a qualitative change in all directions of the methodology for growing crops. The focus is shifted to the modernization and the specification of the activities, to the autonomous detection of problems, to environmentally controlled fight and limitation of environmental damages [3]. Essential features of the smart agriculture are the personalized monitoring and the selective service.

The automated collection of crop data is possible thanks to unmanned aircrafts. Machine learning methods, however, allow different objects in the field to be recognized. The computer vision allows the mapping of the area under cultivation and the specification of the chemical spray operations.

2. Weed Detection Solutions

There are various projects and solutions that implement the monitoring, analysis and planning of the work in agriculture using innovative technologies, including image recognition or recognition of objects in an image.

The reference [4] proposes a method for recognizing chamomile and carrots, which takes into account the overlap of leaves as they grow and divides the mechanism of identification into three stages: image segmentation, feature extraction and classification. The individual steps use K-Means Clustering for splitting the data, Histogram of Oriented Gradient for locating weed regions, and Support Vector Machine for classifying. The method achieves 92% accuracy. In [5] the Convolutional Neural Network detects weeds in images of wheat and rye. The network is trained with over 13,000 annotated high-resolution RGB images. The network achieves an accuracy of 76%. The system in [6] uses bots to detect and to selectively remove weeds in fields of lettuce, cotton and soybean. The system provides 90% lower consumption of chemicals. [7] compares VGGNet, AlexNet, GoogleNet, and DetectNet architectures for detecting three types of weeds (dandelion, ground ivy, spotted spurge) growing among perennial ryegrass. After training with 33,000 examples, VGGNet and DetectNet achieve the highest accuracy – 99% and 98% respectively. The bot in [8] recognizes images using Bayesian algorithms. Thus it has the capacity of planting seeds, watering plants and removing weeds.

According to the accomplished literary reference, it became clear that there is a strong research interest in the field of weed detection among different crop cultures, the main methodology used to be the recognition by machine learning. The present study provides a prototype for the detection of some common and economically important weeds in wheat crops. For this purpose, a deep convolutional neural network has been used.

3. Image dataset

The plant recognition requires a database with a certain number of images of the species required for the neural network training. For the purposes of our assignment, the database should contain images of wheat and weeds at the earliest stage of development. A prerequisite for the good network training is the base to be balanced, diverse and large enough. Creating such a base is not impossible, but there are certain limitations in terms of time interval, meteorological conditions and technique. An alternative is to use a ready, publicly accessible base, despite the trade-offs in terms of some requirements. This study has used the public V2 Plant Seedlings Dataset, created by a team of university researchers in Denmark [9]. The database contains 5539 full-color images of 12 species of plants. These are common crop and weed species in the Danish agriculture. The images are of different sizes.

Several limitations have necessitated the reduction of the original data set. Firstly, only a few of the weeds develop in wheat crops in Bulgaria. For this reason, only 7 plant species are left in the database: common wheat, charlock, loose silky-bent, common chickweed, scentless mayweed, cleavers and blackgrass (Fig. 1).

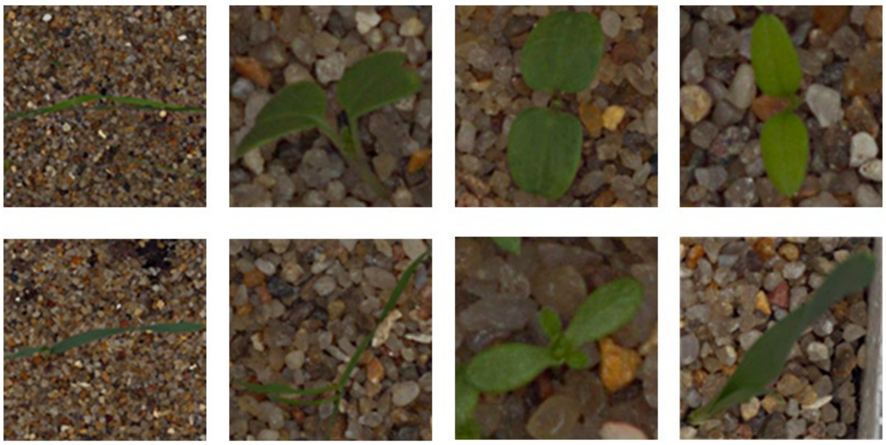


Fig. 1. Wheat-Weed Dataset

Secondly, some of the images in the original database are with very low resolution and are highly defocused. They have been removed. Thus, the final Wheat-Weed Dataset contains 2234 labeled RGB images belonging to 7 classes. These data are divided into three sets – training, validation and testing, in an approximate ratio of 5: 2: 3.

4. Deep Convolutional Neural Network

Convolutional Neural Network (CNN) is the most successful image recognition network. Its main feature is its ability to encode different image characteristics in different layers of its architecture. The combinations between the basic layers (convolution, pooling, fully connected) allow for a wide variety and an application in different domains and for different tasks.

The CNN input layer is a tensor whose dimension depends on the size of the input image. It is passed through a series of layers. The first layers are convolutional. They memorize the elemental characteristics of the image by encoding them into feature maps. Linear filters are used for creating each feature map (Fig. 2).

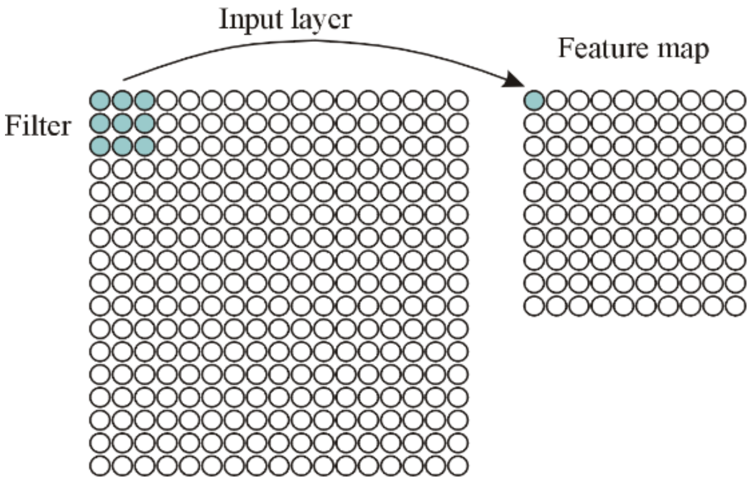


Fig. 2. Creating a feature map in convolutional layer

If the image is full colored, the number of the filters and the characteristic maps is tripled (Fig. 3).

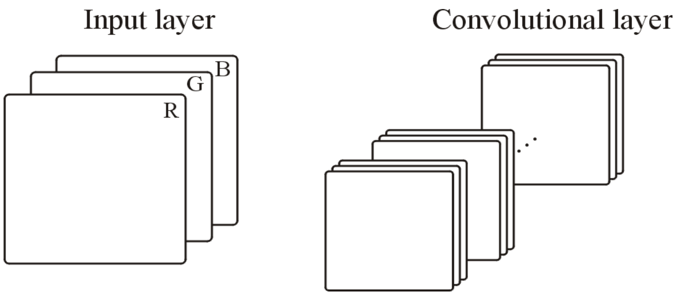


Fig. 3. Convolutional layer (depth = 3)

Filters preserve weights and bias coefficients, and an activation function is applied at the output of neurons (e.g. Sigmoid, Hyperbolic tangent (Tanh), Rectified Linear Unit (ReLU) or its modification [10]). ReLU is one of the most used functions because it retains the gradient speed. On the other hand, it reduces the dependence between the characteristics by excluding the weak ones.

In order to unify the elementary properties, pooling layers are included in the network architecture. They reduce computational resources by reducing the dimension of the data and the amount of parameters that are calculated.

The last layers of the architecture are fully connected. They integrate all the features of the image. For multi-classification problems, the softmax activation function is most often applied at the output. The function generates a vector whose components describe the probable classes.

The capacity of neural networks (the number of features they learn) affects their quality of training. At high capacity, the network may be overfitted. Deep networks are particularly prone to adapting to data. The capacity reduction is accomplished by excluding neurons during the training stages. However, with low capacity, the network may remain underfitted. The prevention of underfitting is based on training with many and varied data submitted in several epochs. A well-trained network indicator, i. e. a network that will be able to generalize, is the minimum difference in accuracy on the training set and validation set.

5. Transfer Learning

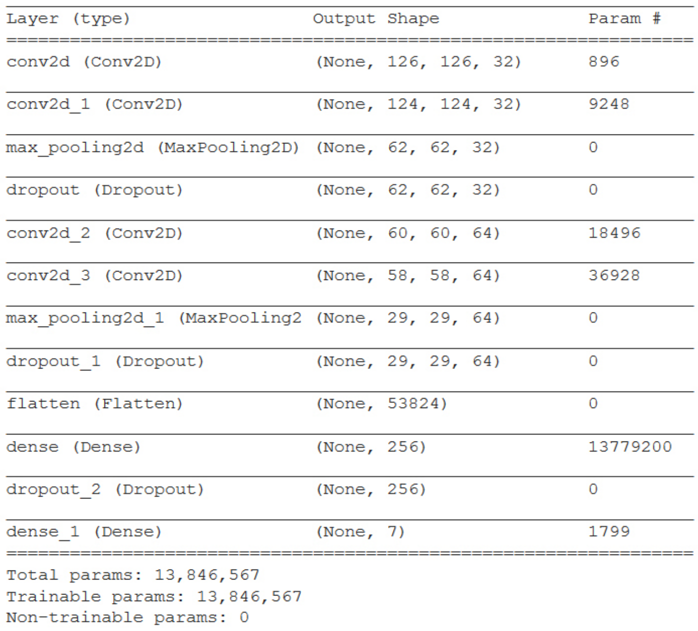
It is known that there is no familiar universal algorithm for the selection of hyperparameters of neural networks. It is also known that their training takes considerable time and computational resources. The transfer training is a good practice in order to reduce these resources – the use of the architecture and/or the weights of an already trained model [11, 12]. Depending on the size of the new training set and on its resemblance to the old one, the pre-trained model is modified in a different way – by removing the classification layer (feature extraction), by keeping the weights in the first few layers (freeze layers), or by retaining only the architecture (using of architecture).

In order to use transfer learning in our case, we need to look for a model trained with data similar to ours. Many DCNN models are known to be trained on huge databases (for large scale image recognition) and achieve a high accuracy (ImageNet [13], VGGNet [14], ResNet [15], Inception [16], Xception [17], etc.). Unfortunately, none of these models has been trained on early-stage plant imaging. In this case two scenarios are possible - to train a new model from scratch or to use partially (by freeze layers method) any ready-made architecture. The only condition for executing the second scenario is that the images have to be the same size as the size of the input layer of the trained model. However, image resolution affects the performance of the network and the used resources. The resolution must be such as to disregard as little useful information as possible, to lead to a reasonable training time and not to exceed the available memory.

6. Realization

For the current experiment was selected the scenario for training a model from scratch. For this purpose, Python 3 and Google's Colaboratory (Colab) platform were used [18]. Colab provides Jupyter notebook service and parallel computing resources needed to train machine learning (ML) models. The software includes the TensorFlow 2.2 framework [19] and Keras 2.3 framework [20]. TensorFlow is a machine learning library and Keras is a neural network library that runs over TensorFlow. For recording the model were used the HDF5 (Hierarchical Data Format) library and the Python interface with it – h5py.

One of the options for creating a model in Keras is it to be created as an object of type *Sequential*, which is a linear stack of layers. The first layer of the sequential model should receive information about the input form of the data. The small amount of training data that we are in disposition of (1236 images) necessitates the use of a slightly higher resolution (128x128), which will allow for the accurate detection of fine details. Deep learning models are taught through batch feeds. The batch is a part of the database that is fed to the network in one iteration. The choice of a batch size (number of examples) is individual and depends mostly on the memory constraints. In our case is selected size 32. Thus the selected resolution and batch size determine the shape of the input tensor: (32, 128, 128, 3), with the last element referring to the number of the color channels in the image. The hidden layers use the form of the output data from the previous layers. The created model operates with 12 hidden layers that process almost 14 million parameters (Fig. 4).



```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 126, 126, 32)      896
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 124, 124, 32)      9248
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 62, 62, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 62, 62, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 60, 60, 64)        18496
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 58, 58, 64)        36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 29, 29, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 29, 29, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 53824)             0
_________________________________________________________________
dense (Dense)                (None, 256)               13779200
_________________________________________________________________
dropout_2 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 1799
=================================================================
Total params: 13,846,567
Trainable params: 13,846,567
Non-trainable params: 0
```

Fig. 4. Selected network architecture

The architecture presented includes four Conv2Ds, two MaxPooling2Ds, and two Dense layers. Three Dropout layers with different sensitivity are included for the prevention of overfitting. ReLu has been used as an activation function for all hidden layers. The Flatten layer changes the dimension of the vectors – from 2D to 1D. The last fully connected layer has seven neurons pertaining to the seven output classes. The activation function in this layer is linear.

Three parameters have to be defined in order to configure the learning process – optimizer, loss-function and accuracy measure. As an optimizer was chosen Adam because it adaptively changes the learning speed and is one of the most effective optimizers. The categorical cross entropy has been chosen as a loss-function and the categorical accuracy – as the measure of accuracy. Both parameters are suitable for a classification of many classes.

In order to diversify the training data, several functions have been applied to it – rotation, shift, flip and zoom. Above this the data has been submitted repeatedly to the network, with total number of epochs being 150 and each epoch being broken into 38 batches.

The training results are shown in Fig. 5. As it can be seen the accuracy of the network (*categorical_accuracy*) increases with each subsequent epoch. The important thing in the case is that the accuracy of metadata (*val_categorical_accuracy*) is also increasing.

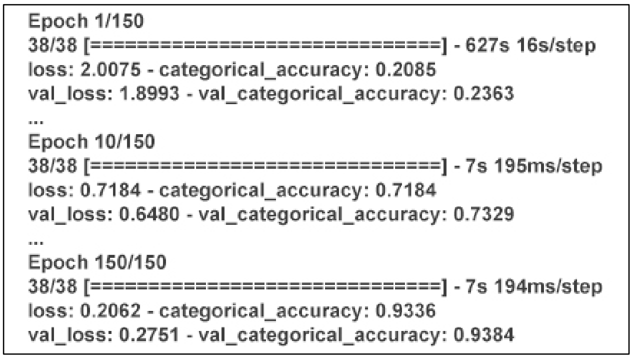

```
Epoch 1/150
38/38 [==============================] - 627s 16s/step
loss: 2.0075 - categorical_accuracy: 0.2085
val_loss: 1.8993 - val_categorical_accuracy: 0.2363
...
Epoch 10/150
38/38 [==============================] - 7s 195ms/step
loss: 0.7184 - categorical_accuracy: 0.7184
val_loss: 0.6480 - val_categorical_accuracy: 0.7329
...
Epoch 150/150
38/38 [==============================] - 7s 194ms/step
loss: 0.2062 - categorical_accuracy: 0.9336
val_loss: 0.2751 - val_categorical_accuracy: 0.9384
```

Fig. 5. Changing the parameters during training

The results obtained at each step in the training step are recorded in the history variable. Thanks to the history, the evolution of accuracy on both sets – training (with accuracy) and validation (with *val_accuracy*) can be visualized (Fig. 6). The overlap of the two graphs shows that the created model is well trained.

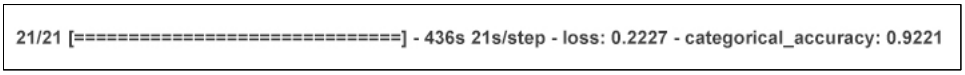

```
21/21 [==============================] - 436s 21s/step - loss: 0.2227 - categorical_accuracy: 0.9221
```

Fig. 6. Validation of accuracy

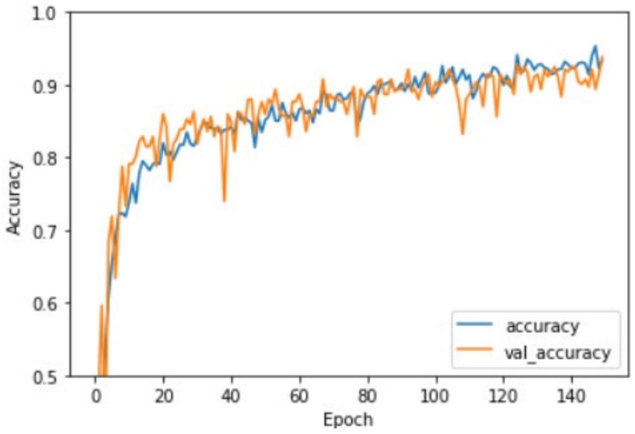
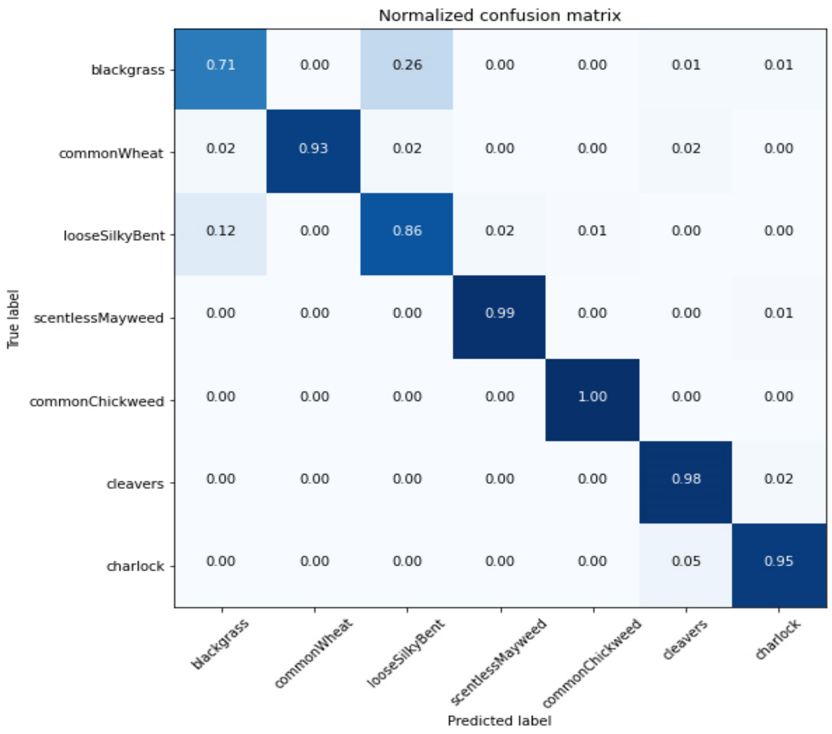


Fig. 7. Accuracy on test data

The overall performance of the ML model can be seen more clearly with the help of a confusion matrix, which shows the accuracy of the network, divided by classes (Fig. 8).



	blackgrass	commonWheat	looseSilkyBent	scentlessMayweed	commonChickweed	cleavers	charlock
blackgrass	0.71	0.00	0.26	0.00	0.00	0.01	0.01
commonWheat	0.02	0.93	0.02	0.00	0.00	0.02	0.00
looseSilkyBent	0.12	0.00	0.86	0.02	0.01	0.00	0.00
scentlessMayweed	0.00	0.00	0.00	0.99	0.00	0.00	0.01
commonChickweed	0.00	0.00	0.00	0.00	1.00	0.00	0.00
cleavers	0.00	0.00	0.00	0.00	0.00	0.98	0.02
charlock	0.00	0.00	0.00	0.00	0.00	0.05	0.95

Fig. 8. Confusion matrix

Rows show the plant species that are fed to the model input. Columns show as what kind has been recognized the plant. The diagonal of the matrix is the most essential. It contains the percentages of the correctly recognized plants. What is noticeable is that the net confuses two of the plants - black grass and loose silky-bent. The recognition rate is below 90%. The reason is that these two species are very similar, and the training data is too small. For other plants the accuracy is higher (over 93%), with common chickweed recognition being 100%.

7. Conclusion

The presented model has been designed in order to be able to recognize weeds in wheat plantations. The model is based on DCNN, Python, TensorFlow and Keras. For training, validation and testing was used Wheat-Weed Dataset, which was processed and reduced from V2 Plant Seedlings Dataset. The database contains 2234 RGB images, divided into 7 classes. The achieved accuracy is 92%.

This model could certainly be improved in the future. But even with this recognition accuracy, it would speed up the process of mapping weeds in wheat crops. The ability to locate different types of weeds will help to delimited treatment with the right herbicides. The future challenge is to determine the leaf mass of weed associations, which will allow the right amount of chemicals to be applied and limit the adverse effects on the environment. The model can also be expanded to identify different types of diseases in wheat plantations, as well as to detect some pests on them. Early diagnosis of diseased plants will reduce producers' losses. A large data set is required to solve all these tasks. As part of our future work is the creation of such a base, with opportunities for continuous expansion.

Acknowledgment

The research is supported by the project ZFTK 9 “Creation of new high-yielding sorts of ordinary winter wheat with a complex of economic qualities and high ecological plasticity, suitable for changing climatic conditions and technological solutions”.

References

1. Ministry of Agriculture and Food, National Plant Protection Service: Guidelines for Integrated Pest Management on Cereals. Sofia (2008). (Министерство на земеделието и храните, Национална служба за растителна защита: Ръководство за интегрирано управление на вредителите по зърнено-житните култури. София (2008).)
2. Ministry of Agriculture and Food, National Plant Protection Service: Methodology for Reporting and Filing of Weeding in Main Field Crops. Sofia (2004). (Министерство на земеделието и храните, Национална служба за растителна защита: Методика за отчитане и картотекиране на заплевеляването при основни полски култури. София (2004).)
3. Ministry of Economy of the Republic of Bulgaria: Concept for digital transformation of Bulgarian industry (Industry 4.0). Available: https://mi.government.bg/en/themes/koncepciya-za-cifrova-transformaciya-na-balgarskata-industriya-industriya-4-0-1862-468.html. (Министерство на икономиката на РБ: Концепция за цифрова трансформация на българската индустрия

(Индустрия 4.0). Available: https://mi.government.bg/bg/themes/koncepciya-za-cifrova-transformaciya-na-balgarskata-industriya-industriya-4-0-1862-468.html.)

4. Saha, D., Hamer, G., Lee, J. Y.: Development of inter-leaves weed and plant regions identification algorithm using histogram of oriented gradient and k-means clustering. RACS '17, Krakow, Poland, DOI: 10.1145/3129676.3129700, (2017).
5. Dyrmann, M., Skovsen, S., Laursen, M., Nyholm, J.: Using a fully convolutional neural network for detecting locations of weeds in images from cereal fields. In: Proc. of 14th International Conference on Precision Agriculture, Montreal, Canada (2018).
6. Blue River Company, Blue River Technology. Available: http://www.bluerivertechnology.com/.
7. Yu, J., Schumann, A., Cao, Z., Sharpe, S., Boyd, N.: Weed detection in perennial ryegrass with deep learning convolutional neural network. Frontiers in Plant Science 10:1422, doi: 10.3389/fpls.2019.01422 (2019).
8. Joshi, J., Samineni, R., Rahul, S., Polepally, S., Kumar, P., Sumedh, K., Tej, D., Rajapriya, V.: Machine learning based cloud integrated farming. In: Proc. of Int. Conference on Machine Learning and Soft Computing, ACM DL, Ho Chi Minh City, Vietnam, https://doi.org/10.1145/3036290.3036297 (2017).
9. Giselsson, T., Jørgensen, R., Jensen, P., Dyrmann, M., Midtiby, H.: A public image database for benchmark of plant seedling classification algorithms. Computer Science: Computer Vision and Pattern Recognition, arXiv: 1711.05458 [cs.CV] (2017).
10. Ide, H., Kurita, T.: Improvement of learning for CNN with ReLU activation by sparse regularization. In: Proc. of International Joint Conference on Neural Networks (IJCNN 2017), DOI: 10.1109/IJCNN.2017.7966185 (2017).
11. Gupta, D.: Transfer learning and the art of using Pre-trained models in deep learning. Analytics Vidhya, Available: https://www.analyticsvidhya.com/blog/2017/06/transfer-learning-the-art-of-fine-tuning-a-pre-trained-model/.
12. Kornblith, S., Shlens, J., Le, Q.V.: Do better imageNet models transfer better. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, California, pp. 2661-2671 (2019).
13. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. CVPR09 (2009).
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Computer Science: Computer Vision and Pattern Recognition, arXiv: 1409.1556v6 [cs.CV] (2015).
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. Computer Science: Computer Vision and Pattern Recognition, arXiv: 1512.03385 (2015).
16. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. Computer Science: Computer Vision and Pattern Recognition, arXiv: 1409.4842 [cs.CV] (2014).
17. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. Computer Science: Computer Vision and Pattern Recognition, arXiv: 1610.02357 (2017).
18. Google Research: Colaboratory. Available: https://colab.research.google.com/notebooks/intro.ipynb.
19. Google Research: TensorFlow Core v2.2.0-rc1. Available: https://www.tensorflow.org/versions/r2.2/api_docs/python/tf.
20. Chollet, F.: Keras: The Python Deep Learning library. Available: https://keras.io/.