Image Semantic Segmentation Based on Convolutional Neural Networks for Monitoring Agricultural Vegetation

Valentin Ganchenko(✉) and Alexander Doudkin

United Institute of Informatics Problems, Surganov st., 6, 220012 Minsk, Belarus
ganchenko@lsi.bas-net.by, doudkin@newman.bas-net.by

Abstract. This paper considers problem of recognition agricultural vegetation state from aerial photographs of various spatial resolutions. Semantic segmentation based on convolutional neural networks is used as a basis for recognition. Two neural networks with SegNet and U-Net architectures are presented and investigated for this aim.

Keywords: Convolutional neural network · Semantic segmentation · Aerial photograph · Agricultural vegetation

1 Introduction

Precision farming implies availability of accurate and promptly updated information about vegetation and soil state. It is possible to obtain such information when remote sensing is used. Remote sensing methods for monitoring agricultural fields make a possibility to quickly identify vegetation areas affected by some diseases. Detection of the diseased areas in early stages of development allows locating and curing the disease promptly and at minimal cost. There are two main approaches to solve the problem of identifying diseased areas - spectrometric and optical [1–7]. The spectrometric approach allows determining many diseases in early stages of development. However, this approach requires multispectral imaging equipment, which is not always possible. In this point of view optical methods are more preferable.

Unmanned aerial vehicles (UAVs) are effective tools of data collection in agriculture because they are cheaper and more efficient in comparison with satellites [8, 9]. UAVs provide visual information about large areas of crops as quickly as possible. Obtained images can import into a GIS database for further processing and analysis, which allows farm managers to make operational decisions.

Convolution neural networks (CNNs) are successfully used for processing of aerial photographs of vegetation in solving various problems of precision farming [10]. In works [11–13], weed extraction in fields with accuracy of more than 90% is shown on

The work was partially supported by Belarusian Republican Foundation for Fundamental Research (project No. Ф18В-005) and the State Committee on Science and Technology of the Republic of Belarus (project no Ф18ПЛШГ-008П).

S. V. Ablameyko et al. (Eds.): PRIP 2019, CCIS 1055, pp. 52–63, 2019.
https://doi.org/10.1007/978-3-030-35430-5_5

data obtained from a robot, where CNN is used for classification of objects and semantic segmentation. Residual CNN is used for semantic segmentation to detect flowers in task of estimating flowering intensity to predict yield [14]. At the same time, detection accuracy is achieved 67–94%, depending on photographed plants. The yield is also estimated for the already growing fruits [15], for which multi-layer perceptron and CNN are used. In [16], CNN model is presented for extracting vegetation from Gaofen-2 remote sensing images. The authors have created two-layer encoder based on CNN, that allows to obtain of 89–90% accuracy of identification. The first layer has two sets of convolutional kernels for selection of features of farmland and woodlands, respectively. The second level consists of two coders that use nonlinear functions to encode the features and to compare codes with corresponding category number. CNNs also can be applied for damage degree evaluation of individual plants. So in [17] U-Net scheme is used, a damage degree of cucumber foliage by powdery mildew is estimated to within 96%. Based on CNN semantic segmentation is also used for thematic mapping. For example, it was shown in [18], where vegetative cover for agricultural land is assessed.

The presented work focuses on recognition of areas of vegetation, state of which has changed due to influence of disease. Two CNNs for implementing of semantic segmentation of color images of agricultural fields is proposed. In this case, disease classification is not performed at this stage. The aim of the work is to develop algorithms for processing of digital color images of various spatial resolutions.

2 Formulation of Problem

Task of the research is to develop transformation algorithm $A : I_{orig} \rightarrow I_{result}$, which allows to obtain image I_{result} from original image of agricultural field I_{orig}. Each pixel $I_{orig}(x, y)$ is a point in RGB space and each pixel $I_{result}(x, y)$ corresponds to one of four classes (“soil”, “healthy vegetation”, “diseased vegetation” and “other objects”).

Materials for research are photographs both of individual plants and an experimental potato field. The pictures were made from a height of 5, 15, 50, and 100 m [19, 20]. To obtain data, small parts of the field were selected using four square marks. The length of the side of the square is one meter; the width of the two black lines is 20 cm (Fig. 1). The marks allow not only to determine area for research, but also to calculate image spatial resolution.

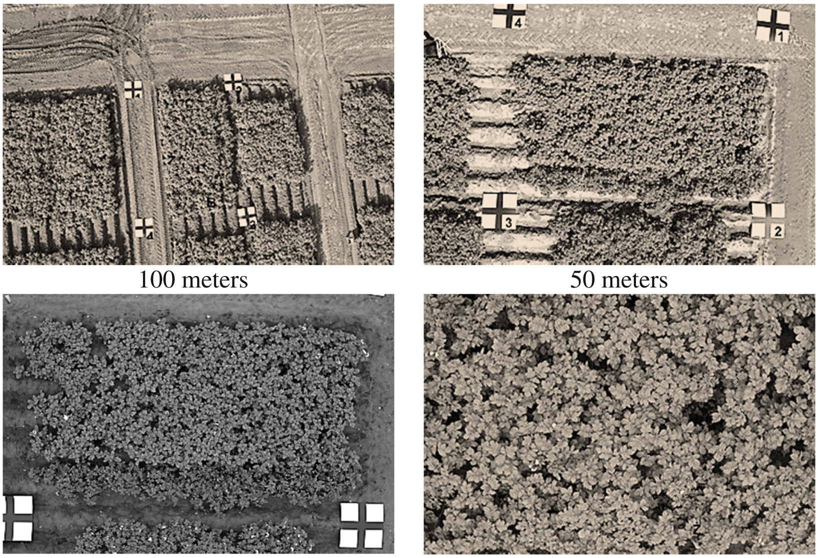
Three groups of plants are observed:

- plants infected with the disease alternaria;
- plants infected with bacterial disease erwinia;
- healthy plants (control group).

The plants were photographed daily at 8, 10, 12, 14 and 16 h during the 8 days in July.

As a result of the diseases mentioned above, chlorophyll is destroyed in potato leaves, what leads to a change in color of plants. Also it should be noted that in clear weather, the sun’s glare on leaves also creates yellow effect, what introduces an additional error during automatic processing.

Histogram analysis of color characteristics of various types of photographs shows a noticeable difference between images of soil and vegetation, as well as the difference in blue channel for healthy and disease plants. For example, for the images of healthy, diseased vegetation and soil in respective histograms, it is visible that the histograms for soil are different from histograms for vegetation on each color channel, and histograms for healthy and diseased vegetation channels differ in shape (Fig. 2).



100 meters

50 meters

15 meters

5 meters

Fig. 1. Samples of origin images

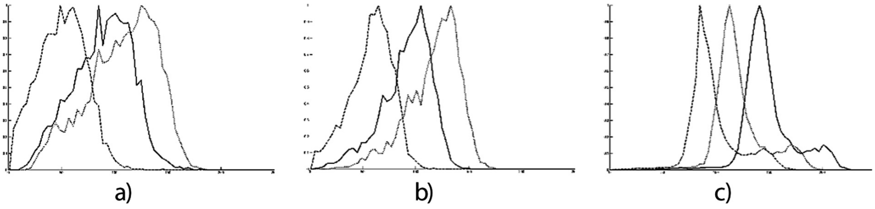


Fig. 2. Histograms: (a) diseased plants; (b) healthy plants; (c) soil

However, presence of several type objects in the selected areas of the images leads to distortion of histogram of the objects – bins will be shifted and there won't be clear peaks. Such distortions, as well as a significant similarity of color characteristics of healthy and diseased vegetation, require information about structure of images of various classes for their recognition. Structural information can be taken into account when CNNs are used as the basis for the proposed algorithms.

3 Preparing of Data for Training and Validation

The training set was obtained by “slicing” existing aerial photographs with labeled areas. At the same time, sections of 256 × 256 pixels were cut with overlapping, vertical and horizontal reflection, as well as with the addition of turns at angles multiple of 90°. A class mask is a halftone image that has the same size as the image. A mask image contains the number of brightness levels which equals to the number of the classes in the image. The following brightness values correspond to the classes: 0 – “soil”, 1 – “healthy vegetation”, 2 – “diseased vegetation”, 3 – “other objects”.

4 Based on SegNet Segmentation

It is proposed the CNN based on SegNet architecture [21, 22] (denote it by A_s; view of this architecture is presented on Fig. 3) that segments images into four segments: “soil”, “healthy vegetation”, “diseased vegetation” and “other objects”.

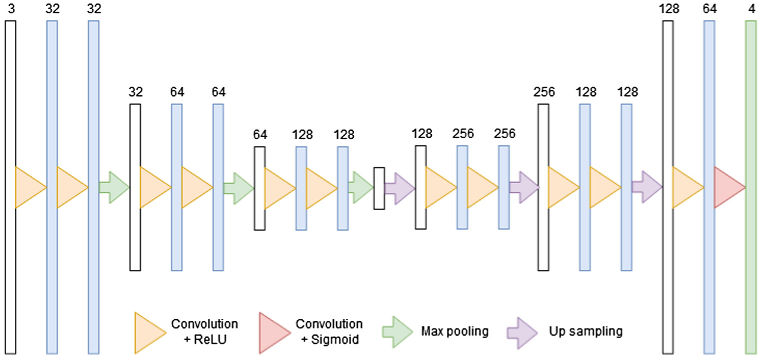


Fig. 3. Implemented SegNet architecture

Empirically selected following parameters of the CNN:

- Input layer size: 256 × 256 × 3 (color image).
- Convolutional layer Conv2D_1.1: filter size F_s = 3, filters count F_c = 32, activation function – ReLU.
- Convolutional layer Conv2D_1.2: filter size F_s = 3, filters count F_c = 32, activation function – ReLU.
- Max pooling layer MaxPooling2D_1: filter size F_s = 2.
- Convolutional layer Conv2D_2.1: filter size F_s = 3, filters count F_c = 64, activation function – ReLU.
- Convolutional layer Conv2D_2.2: filter size F_s = 3, filters count F_c = 64, activation function – ReLU.
- Max pooling layer MaxPooling2D_2: filter size F_s = 2.

- Convolutional layer Conv2D_3.1: filter size $F_s = 3$, filters count $F_c = 128$, activation function – ReLU.
- Convolutional layer Conv2D_3.2: filter size $F_s = 3$, filters count $F_c = 128$, activation function – ReLU.
- Max pooling layer MaxPooling2D_3: filter size $F_s = 2$.
- Upsampling layer UpSampling2D_1: scale factor = 2 interpolation – bilinear.
- Convolutional layer Conv2D_4.1: filter size $F_s = 3$, filters count $F_c = 256$, activation function – ReLU.
- Convolutional layer Conv2D_4.2: filter size $F_s = 3$, filters count $F_c = 256$, activation function – ReLU.
- Upsampling layer UpSampling2D_2: scale factor = 2 interpolation – bilinear.
- Convolutional layer Conv2D_5.1: filter size $F_s = 3$, filters count $F_c = 128$, activation function – ReLU.
- Convolutional layer Conv2D_5.2: filter size $F_s = 3$, filters count $F_c = 128$, activation function – ReLU.
- Upsampling layer UpSampling2D_3: scale factor = 2 interpolation – bilinear.
- Convolutional layer Conv2D_6.1: filter size $F_s = 3$, filters count $F_c = 64$, activation function – ReLU.
- Output convolutional layer Conv2D_6.2: filter size $F_s = 3$, filters count $F_c = 4$, activation function – sigmoid, output layer size – 256 × 256 × 4.

Loss function – softmax cross entropy [23].
Training:

- Training set size: 20000 images.
- Validation set size: 4000 images.
- Accuracy for validation set: 92.36%.

5 Based on U-Net Segmentation

The U-Net A_u segmenter is a CNN (Fig. 4), which segments image into four segments: “soil”, “healthy vegetation”, “diseased vegetation” and “other objects”. This architecture differs from SegNet by presence of additional connections between convolution layers, which is technically expressed by the addition of concatenation layers. Empirically selected the following parameters of the CNN:

- Input layer size: 256 × 256 × 3 (color image).
- Convolutional layer Conv2D_1.1: filter size $F_s = 3$, filters count $F_c = 32$, activation function – ReLU.
- Convolutional layer Conv2D_1.2: filter size $F_s = 3$, filters count $F_c = 32$, activation function – ReLU.
- Max pooling layer MaxPooling2D_1: filter size $F_s = 2$.
- Convolutional layer Conv2D_2.1: filter size $F_s = 3$, filters count $F_c = 64$, activation function – ReLU.
- Convolutional layer Conv2D_2.2: filter size $F_s = 3$, filters count $F_c = 64$, activation function – ReLU.

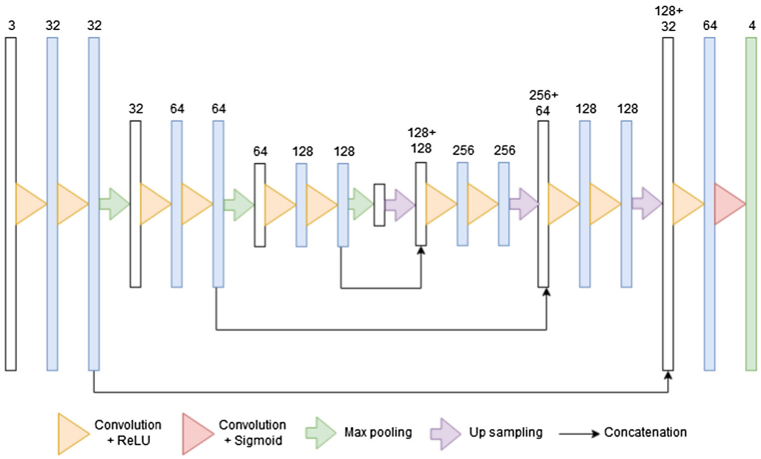


Fig. 4. Implemented U-Net architecture

- Max pooling layer MaxPooling2D_2: filter size $F_s = 2$.
- Convolutional layer Conv2D_3.1: filter size $F_s = 3$, filters count $F_c = 128$, activation function – ReLU.
- Convolutional layer Conv2D_3.2: filter size $F_s = 3$, filters count $F_c = 128$, activation function – ReLU.
- Max pooling layer MaxPooling2D_3: filter size $F_s = 2$.
- Upsampling layer UpSampling2D_1: scale factor = 2 interpolation – bilinear.
- Layer for concatenation of UpSampling2D_1 and Conv2D_3.2.
- Convolutional layer Conv2D_4.1: filter size $F_s = 3$, filters count $F_c = 256$, activation function – ReLU.
- Convolutional layer Conv2D_4.2: filter size $F_s = 3$, filters count $F_c = 256$, activation function – ReLU.
- Upsampling layer UpSampling2D_2: scale factor = 2 interpolation – bilinear.
- Layer for concatenation of UpSampling2D_2 and Conv2D_2.2.
- Convolutional layer Conv2D_5.1: filter size $F_s = 3$, filters count $F_c = 128$, activation function – ReLU.
- Convolutional layer Conv2D_5.2: filter size $F_s = 3$, filters count $F_c = 128$, activation function – ReLU.
- Upsampling layer UpSampling2D_3: scale factor = 2 interpolation – bilinear.
- Layer for concatenation of UpSampling2D_3 and Conv2D_1.2.
- Convolutional layer Conv2D_6.1: filter size $F_s = 3$, filters count $F_c = 64$, activation function – ReLU.
- Output convolutional layer Conv2D_6.2: filter size $F_s = 3$, filters count $F_c = 4$, activation function – sigmoid, output layer size – $256 \times 256 \times 4$.

Loss function – softmax cross entropy.

Training:

- Training set size: 20000 images.
- Validation set size: 4000 images.
- Accuracy for validation set: 93.65%.

6 Output Data Structure

The output of implemented CNNs is 256 × 256 × 4 matrix, where the dimensions "256 × 256" correspond to the size of the input image, and "4" – to the number of the required classes: "soil", "healthy vegetation", "diseased vegetation" and "other objects. Thus, the output is four matrices which elements are the values of probability of belonging of pixels of the original image to the particular class. After normalization of the values for each pixel, we obtain a fuzzy value that characterizes belonging of pixel to the desired classes.

7 Recognition Algorithm

In general, the recognition algorithm (transformation $A : I_{orig} \rightarrow I_{result}$) can be represented as follows:

1. Load origin color image I_{orig}.
2. Divide I_{orig} to parts $O_i\left(I_{orig}\right)$ with size 256 × 256. For each part:
   2.1 Copy selected part $O_i\left(I_{orig}\right)$ with size 256 × 256 as color image.
   2.2 Transform obtained image $O_i\left(I_{orig}\right)$ by segmenter $A \in \{A_S, A_u\}$ to matrix $Segm_A$ with size 256 × 256 × 4.
   2.3 Obtain class index for each pixel of the image $O_i\left(I_{orig}\right)(x, y)$: $x \in [0, 255]$, $y \in [0, 255]$:

$$index = argmax([A(x, y)]),$$

   where $Segm_A(x, y)$ – vector with 4 values which correspond to degree of belonging to the required classes of the origin image $O_i\left(I_{orig}\right)$.
   2.4 Set values of the pixels of output image $I_{result}(O_i)$. Each value corresponds to pseudocolor of the class index: black – to soil, dark-gray – to healthy vegetation, light-gray – to diseased vegetation, white – to the other objects.
3. Save the obtained image I_{result}.

8 Testing

Segmenters were tested on validation set. At the same time, accuracy was assessed both for each class separately and for all classes as a whole. The obtained test results are shown in Table 1.

Table 1. Segmenter test results

Classes	Accuracy %	
	SegNet	U-Net
Soil	84.88	87.99
Healthy vegetation	95.41	96.89
Diseased vegetation	77.48	74.56
Other objects	88.80	88.90
Average	**92.36**	**93.65**

Due to the imbalance of classes in the origin data, an additional evaluation is required. The result data are summarized in confusion matrix presented in Table 2. The value in the matrix is given as the ratio of the number of pixels belonging to the class to the total number of pixels of all classes in the sample.

Table 2. Confusion matrix

Predicted classes	Real classes			
	SegNet			
	Soil	Healthy	Diseased	Others
Soil	11.51	0.75	0.20	0.02
Healthy	1.41	73.49	1.79	0.03
Diseased	0.62	2.71	6.88	0.00
Other objects	0.03	0.07	0.00	0.48
	U-Net			
Soil	11.93	0.86	0.31	0.02
Healthy	1.34	74.63	1.95	0.04
Diseased	0.27	1.50	6.62	0.00
Other objects	0.02	0.03	0.00	0.48

To assess quality of the segmentation, corresponding values of precision, recall and F_1-score [24] were calculated (TP – True Positives count, FP – False Positives count, FN – False Negatives count):

$$Precision = \frac{TP}{TP + FP}, \; Recall = \frac{TP}{TP + FN}, \; F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall},$$

Values of these measures are presented in Table 3.

Table 3. Precision, recall and F1-score

Classes	SegNet		
	Precision	Recall	F_1
Soil	0.92	0.85	0.88
Healthy	0.96	0.95	0.96
Diseased	0.67	0.77	0.72
Others	0.83	0.89	0.86
	U-Net		
Soil	0.91	0.88	0.89
Healthy	0.96	0.97	0.96
Diseased	0.79	0.75	0.77
Others	0.90	0.89	0.89

The greatest number of errors occurred in areas that correspond to boundary of healthy vegetation and soil (especially in places where small areas of soil are surrounded by vegetation, what which creates a shadow on this area of soil).

Additionally, Table 4 provides estimations of the number of errors for each class separately. It can be seen that the significant number of errors occurs when the soil is not correctly identified as healthy vegetation (boundaries of vegetation and soil, small patches of soil among vegetation). The greatest number of errors occurs when diseased areas of vegetation are classified as healthy on any image parts where signs of damage are not sufficiently pronounced.

Figure 5 shows an example of the original image part and the corresponding class labels.

Table 4. Error estimation

Predicted classes	Error %			
	SegNet			
	Soil	Healthy	Diseased	Others
Soil	–	0.98	2.31	4.67
Healthy	10.38	–	20.21	6.53
Diseased	4.54	3.52	–	0.01
Other objects	0.2	0.09	0	–
	U-Net			
Soil	–	1.12	3.49	4.28
Healthy	9.85	–	21.96	6.82
Diseased	2.01	1.95	–	0
Other objects	0.15	0.04	0	–

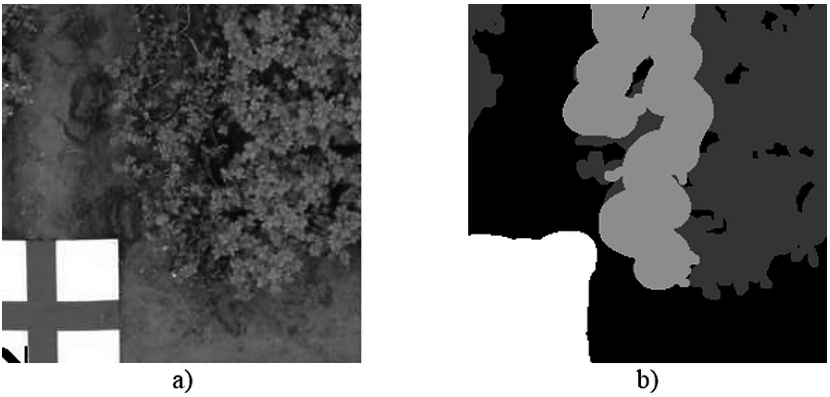


Fig. 5. Example of original aerial image (a) and corresponding labeled classes (b)

Figure 6 shows the classes obtained for this image part. For comparison, the classes are also given labeled by an expert.

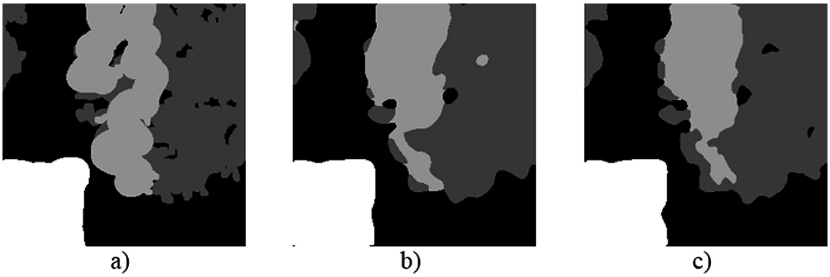


Fig. 6. Labels of classes (a); classes obtained using SegNet (b) and U-Net (c)

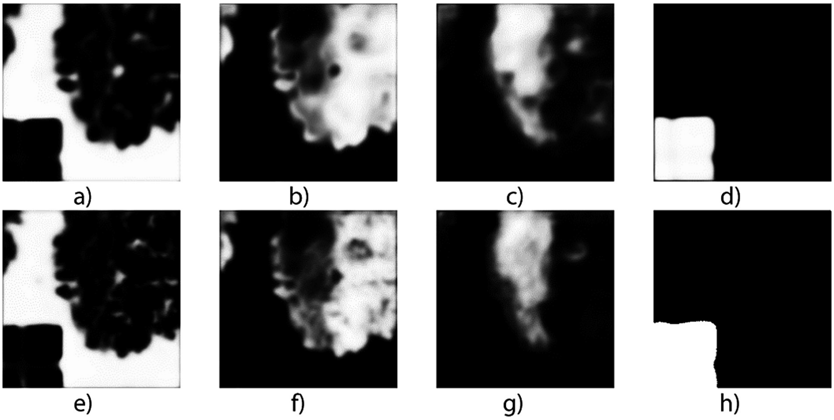


Fig. 7. Degrees of belonging of points of part of a segmented image to classes, obtained using SegNet (a–d) and U-Net (e–h)

Figure 7 shows degrees of belonging of pixels of segmented image to the classes: 7a, 7e – soil, 7b, 7f – healthy vegetation, 7c, 7 g – diseased vegetation, 7d, 7 h – other objects.

9 Conclusions

Semantic segmenters for processing of aerial photographs of agricultural fields were proposed and implemented using the Keras library (the Tensorflow library was used as the backend). The segmenters are built on SegNet and U-Net architectures and trained for obtaining the four classes: “soil”, “healthy vegetation”, “diseased vegetation” and “other objects”. Using the proposed segmenters, it was possible to achieve an accuracy of 92–93%. In this case, the greatest number of errors occurs for diseased vegetation, which can be mistakenly attributed to healthy in the case of small damaged areas, as well as in cases when significantly diseased plants are interspersed with healthy, as well as soil plots.

Further research suggests to reduce errors in problem areas.

References

1. Belyayev, B.I., Katkovskiy, L.V.: Optical remote sensing, 455 p. BSU, Minsk (2006). [in Russian]
2. Schowengerdt, R.A.: Remote Sensing. Models and Methods for Image Processing, 3rd edn, 558 p. Academic Press (2007)
3. Chao, K., Chen, Y.R., Kim, M.S.: Machine vision technology for agricultural applications. Trans. Comput. Electron. Agric. **36**, 173–191 (2002). Elsevier science
4. Kumar, N., et al.: Do leaf surface’ characteristics affect agrobacterium infection in tea [camellia sinensis (1.)]. J. Biosci. **29**(3), 309–317 (2004)
5. Wu, L., et al.: Identification of weed, corn using BP network based on wavelet features and fractal dimension. Sci. Res. Essay **4**(11), 1194–1400 (2009)
6. Qin, Z., Zhang, M.: Detection of rice sheath blight for in-season disease management using multispectral remote sensing. Int. J. Appl. Earth Obs. Geoinf. **7**, 115–148 (2005)
7. Aksoy, S., Akcay, H.G., Wassenaar, T.: Automatic mapping of linear woody vegetation features in agricultural landscapes using very high-resolution imagery. IEEE Trans. Geosci. Remote Sens. **48**(1, 2), 511–522 (2010)
8. Abdullahi, H.S., Zubair, O.M.: Advances of image processing in precision agriculture: using deep learning convolution neural network for soil nutrient classification. J. Multidisciplinary Eng. Sci. Technol. (JMEST) **4**(8), 7981–7987 (2017)
9. Wright, D., Rasmussen, V., Ramsey, R., Baker, D., Ellsworth, J.: Canopy reflectance estimation of wheat nitrogen content for grain protein management. GISci. Remote Sens. **41** (4), 287–300 (2004)
10. Khobragade, A., Pooja, M.G., Singh, R.K.: Feature extraction algorithm for estimation of agriculture acreage from remote sensing images, pp. 5–9 (2016)
11. Huang, H., Deng, J., Lan, Y., Yang, A., Deng, X., Zhang, L.: A fully convolutional network for weed mapping of unmanned aerial vehicle (UAV) imagery. PLoS ONE **13**(4), e0196302 (2018)

12. Sa, I., et al.: weedNet: dense semantic weed classification using multispectral images and MAV for smart farming. IEEE Robot. Autom. Lett. **3**(1), 588–595 (2018)
13. Potena, C., Nardi, D., Pretto, A.: Fast and accurate crop and weed identification with summarized train sets for precision agriculture. In: Chen, W., Hosoda, K., Menegatti, E., Shimizu, M., Wang, H. (eds.) IAS 2016. AISC, vol. 531, pp. 105–121. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-48036-7_9
14. Dias, P.A., Tabb, A., Medeiros, H.: Multispecies fruit flower detection using a refined semantic segmentation network. IEEE Robot. Autom. Lett. **3**(4), 3003–3010 (2018)
15. Bargoti, S., Underwood, J.P.: Image segmentation for fruit detection and yield estimation in apple orchards. J. Field Robot. **34**(6), 1039–1060 (2017)
16. Zhang, C., et al.: Segmentation model based on convolutional neural networks for extracting vegetation from Gaofen-2 images. J. Appl. Remote Sens. **12**(4), 042804 (2018)
17. Lin, K., Gong, L., Huang, Y., Liu, C., Pan, J.: Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network. J. Front. Plant Sci. **10**, 10 p (2019). Article 155
18. Xu, L., Ming, D., Zhou, W., Bao, H., Chen, Y., Ling, X.: Farmland extraction from high spatial resolution remote sensing images based on stratified scale pre-estimation. J. Remote Sens. **11**(2), 10–19 (2019)
19. Sobkowiak, B., et al.: Zastosowanie technik analizy obrazu do wczesnego wykrywania patogenow ziemniaka. Praca nie publicowana. PIMR, Poznan (2006)
20. Sobkowiak, B., et al.: Zastosowanie technik analizy obrazu do wczesnego wykrywania zarazy ziemnechanej w warynkach polowych. Praca nie publicowana. PIMR, Poznan (2007)
21. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning, 800 p. The MIT Press (2016)
22. Nikolenko, S, Kadurin, A., Archangelskaya, E.: Deep Learning, 480 p. Piter, Saint Petersburg (2018). (in Russian)
23. Tensorflow API documentation. https://www.tensorflow.org/api_docs/python/tf/nn/softmax_cross_entropy_with_logits_v2. Accessed 04 Aug 2019
24. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In: Sattar, A., Kang, B.-H. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 1015–1021. Springer, Heidelberg (2006). https://doi.org/10.1007/11941439_114