Turkish Journal of Agriculture and Forestry

Volume 46 | Number 3 Article 11

1-1-2022

# Dissimilarity-based active learning for embedded weed identification

YANA YANG

YANG LI

JIACHEN YANG

JIABAO WEN

TÜBİTAK

Turkish Journal of Agriculture and Forestry

http://journals.tubitak.gov.tr/agriculture/

Research Article

Turk J Agric For
(2022) 46: 390-401

doi:10.55730/1300-011X.3011

# Dissimilarity-based active learning for embedded weed identification

**Yana YANG[1,2], Yang LI[1,2,*], Jiachen YANG[2], Jiabao WEN[2]**
[1]College of Mechanical and Electrical Engineering, Shihezi University, Shihezi, China
[2]School of Electrical and Information Engineering, Tianjin University, Tianjin, China

**Received:** 01.03.2022 • **Accepted/Published Online:** 22.04.2022 • **Final Version:** 21.06.2022

**Abstract:** Weed identification helps ensure crop yield and realize precision agriculture. Although the deep learning-based methods have achieved high performance, their needed large-scale annotated data is difficult to obtain, and the massive parameters lead to difficulties in model deployment in embedded applications. To develop efficient crop weeds classification system, we propose a dissimilarity-based method to select few but representative samples and consider data diversity. Many experiments are conducted on two different datasets. Results show that the proposed method can obtain test accuracy of 90.75% with 32% amount of the DeepWeeds dataset, up to 99.18% of the baseline accuracy. For the Grass-Broadleaf dataset, our approach can obtain the test accuracy of 98.97% with 27.8% amount of the dataset, up to 99.5% of the baseline accuracy. Further, the designed model is compressed from 117.9 MB to 8.6 MB, with a compression ratio of 92.7%, while with only 1% degradation of accuracy. Finally, the compressed model is deployed to NVIDIA Jetson AGX Xavier, running at 192 fps. This work can lay a foundation for the following research about few-shot learning, data quality assessment, and model deployment in agriculture.

**Key words**: Smart agriculture, data quality, metric, few-shot, compression

## 1. Introduction

Intelligent agriculture is a new development trend of digital agriculture (Jiang et al, 2021; Li and Yang, 2021). The crop weeds classification is increasingly significant in precision agriculture. In traditional agriculture, farmers rely on the manual identification of weeds. However, in a large area of field such as Xinjiang, northwest China, there usually exists a variety of weeds. Many kinds of them have similar characteristics, making manual identification more and more difficult.

Deep Convolutional Neural Networks (CNNs) have attracted much attention from researchers (Nie et al, 2019; Li et al, 2020; Li and Chao, 2021; Li et al, 2022), owing to the excellent feature extraction capability in agricultural classification tasks. The accuracy of classification is constantly improving. Considering the difficulty of manual identification, many researchers have used CNNs for precision agriculture. For instance, rice pests and diseases were classified and a lightweight architecture for embedded devices was developed (Rahman et al, 2020). Several CNNs were compared to identify barley, considering accuracy, computation, and training cost (Kozłowski et al, 2019). A new CNN model was proposed to use many kinds of features for crop classification (Yang et al, 2020). A deep CNN was utilized to recognize cotton boll status, which can substitute manual identification (Li et al, 2020). A deep CNN for accurate crop identification was proposed and tested on the self-built WHU-Hi dataset (Zhong et al, 2020). A CNN model was used to classify many kinds of crops from different source images (Gu et al, 2019). It cannot be denied that CNNs used for precision agriculture have made remarkable achievements over the past several years.

However, there are two main problems with the current deep learning methods. First, deep CNNs based on deep learning require a huge mass of data (Li and Chao, 2021; Li et al, 2021), which may be difficult to obtain in some cases. In addition, when facing large amounts of datasets, labelling them is difficult and costs much of the time. Second, as the depth of the network increases, there are more than millions of parameters in the deep model (He et al, 2019). High-performance servers with GPUs are indispensable. If we want to deploy the deep CNNs for weed classification in the field, it is impractical to use those servers (Villa-Henriksen et al, 2020).

Due to the difficulties of acquiring a large amount of data and labelling, it is necessary to study with a small number of samples. Recently some methods with a few samples have made much progress, which contain few-shot learning (Li and Yang, 2020), transfer learning

* Correspondence: liyang328@shzu.edu.cn

(Mohanty et al, 2016), active learning (Yin et al, 2017; Yoo and Kweon, 2019). On account of numerous parameters of deep CNNs, many researchers have proposed different methods, such as weight quantization (Rastegari et al, 2016), filter pruning (Lin et al, 2020; Yan et al, 2020), and weight decomposition (Li et al, 2019).

In this paper, we introduce a new method for weed classification, named dissimilarity-based active learning (DBAL), which can pick out a few samples yet consider the overall diversity of the dataset. We carry out comparative experiments on two different datasets, DeepWeeds and Grass-Broadleaf datasets for weed recognition. Experiment results show that our method can achieve 90.75% accuracy with 32% amount of the DeepWeeds dataset, up to 99.18% of the baseline accuracy (91.5%). Results can validate the effectiveness and generalization of our DBAL method. To degrade the number of parameters and model size, we prune the unimportant convolutional channels according to scale factors of Batch Normalization (BN) layers. We compress the model from 117.9MB to 8.6MB, up to a compression ratio of 92.7%, with an inconspicuous 1% accuracy drop. Finally, we transplant the compressed model to NVIDIA Jetson AGX Xavier, running at 192fps. This study attempts to contribute to the practicality of precision agriculture. In summary, the main contributions are as follows.

(1) We propose a DBAL method, which can select fewer instances to cover the diversity of the whole dataset as much as possible according to metric.

(2) Experiment results on the two different weed datasets show that our method is effective and robust, also has competitive performance.

(3) By model compression, the model size degrades from 117.9 MB to 8.6 MB, with 1% accuracy drop. The compressed model can run at 192 fps on NVIDIA Jetson AGX Xavier.

## 2. Materials and methods

### 2.1. Dataset

To verify the effectiveness and generalization of our proposed DBAL method, we carry out the experiments on the two different public datasets, Grass-Broadleaf and DeepWeeds datasets.

#### 2.1.1. Grass-Broadleaf dataset

The Grass-Broadleaf dataset consists of 4 classes. Figure 1 shows some examples of the dataset. It belongs to the Kaggle competition, which can be found at the following site (https://www.kaggle.com/fpeccia/weed-detection-in-soybean-crops).

There are 15,336 images in the Grass-Broadleaf dataset. The number of images in the dataset is shown in Table 1.

#### 2.1.2. DeepWeeds dataset

The DeepWeeds dataset contains 9 different kinds (Olsen et al, 2019; Lu and Young, 2020). Figure 2 shows some examples of the dataset. This dataset will help identify various weeds under different growing states from many weeds in the fields.

There are 17,509 images with 256 × 256 pixels in the DeepWeeds dataset. The number of images in the dataset is shown in Table 2. The number of images for each class surpasses 1000.

### 2.2. The overall framework

Benefiting from the development of deep learning, the classification accuracy is significantly improved. However, it often relies on a large-scale dataset. There is no doubt that labelling will be time-consuming in this case. In addition, the network contains a huge number of parameters, it needs high-performance GPUs as support. So, on one hand, we use active learning to find fewer but more useful samples from the large-scale of the dataset. On the other hand, we use model compression to reduce network parameters and model size, making the network more suitable for embedded platforms. Figure 3 shows the overall framework of this study.

When it comes to active learning, we often divide the data into a small labelled pool $L$ and a large unlabelled pool $U$. Only samples in the $L$ will be used to train the classification network. We want to find some samples in $U$ which will help improve the test precision if they are labelled. Then those chosen samples will be added to $L$ for network training and evaluation. This active learning

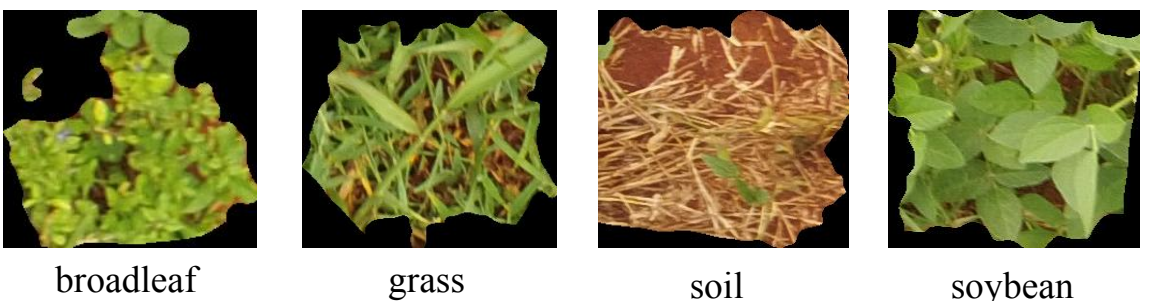


**Figure 1.** Some examples of the Grass-Broadleaf dataset.

**Table 1.** The number of images in the Grass-Broadleaf dataset.

| Index of class | Class name | Number of images |
|---|---|---|
| 0 | broadleaf | 1191 |
| 1 | grass | 3520 |
| 2 | Soil | 3249 |
| 3 | soybean | 7376 |

process can repeat several times until it meets the demand for the number of samples or precision.

We propose an active learning method, for agricultural datasets used for classification, our method can find the samples from the unlabeled pool *U* that are the most dissimilar to those in the labelled pool *L*. Then those chosen samples from *U* are labeled and put into *L* for the next step of training. We can get a high-performance classifier instead of using all the samples of the dataset.

The classification networks based on deep learning usually contain a large number of parameters, which is not affordable for embedded platforms with limited computing resources. Model compression is an effective method to solve the problem. Some convolutional channels in the network are redundant and firstly we should find the least important channels in the network. We choose the scale factor in BN layers to judge the importance of the convolutional channels (Liu et al, 2017). Then we can remove the unimportant channels corresponding to the small-scale factors in the network according to the pruning ratio. Model compression perhaps leads to the drop in accuracy, fine-tuning will help compensate for

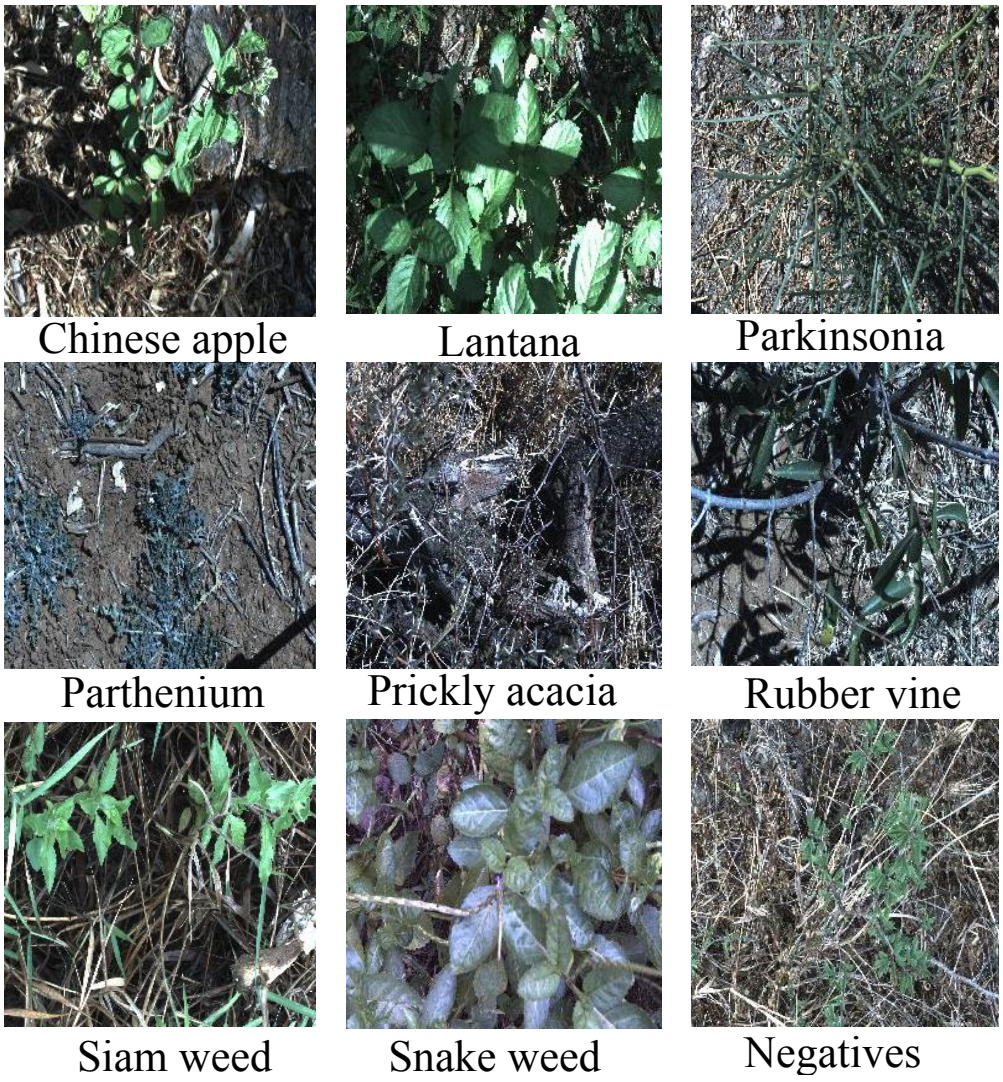


**Figure 2.** Some examples of the DeepWeeds dataset.

**Table 2.** The number of images in the DeepWeeds dataset.

| Index of class | Class name | Number of images |
|---|---|---|
| 0 | Chinese apple | 1125 |
| 1 | Lantana | 1064 |
| 2 | Parkinsonia | 1031 |
| 3 | Parthenium | 1022 |
| 4 | Prickly acacia | 1062 |
| 5 | Rubber vine | 1009 |
| 6 | Siam weed | 1074 |
| 7 | Snake weed | 1016 |
| 8 | Negatives | 9106 |

the accuracy. In the end, we transplant our model to the embedded terminal (such as NVIDIA Jetson AGX Xavier), which will contribute to the practical smart agricultural applications.

### 2.3. Proposed DBAL method

For active learning, we aim to choose some samples from *U* which contain more valuable information if they are labelled and used for training. If these chosen samples are dissimilar to those in the labelled pool *L*, they may contain more useful information. In other words, if these chosen samples are labelled, then *L* will cover the diversity of the dataset as much as possible and help improve the accuracy. In this phase, we do not care about the true label of the samples in *U,* whatever it is lantana or another kind of weed, we just take the dissimilarity between the samples in *U* and those in *L* into consideration.

The dissimilarity-based active learning (DBAL) method is shown in Figure 4.

Firstly, we use a well-trained classification network to extract the feature embedding of the instances in the *L* and *U.* Next, we train a simple binary classification network, which consists of several full connected layers, to distinguish whether the samples belong to *L* or *U.* Then, feature embedding will be extracted again by the binary network. Lastly, K-means is utilized to calculate the cluster centroid of the samples in *L*. Then we use the Euclidean distance to measure the dissimilarity between the cluster centroid and each sample in *U*. We will choose and label those top-K samples of *U* according to the sorted distance, which may contain more valuable information.

In specific, the images in the labelled pool *L* and unlabeled pool *U* are regarded as the input space *X1*. Only images in *L* are used for training a classification network *N1*, which can distinguish several kinds of weeds. Then, we remove the last classification layer of the network *N1* and extract all the features embedding corresponding to the images, which can be marked as feature space *X2*. The size of feature space *X2* is 512-d.

Next, we make a binary dataset and propose a simple binary classification issue, where the feature space *X2* is regarded as the input and the label space *Y* only contains 0 and 1. In the feature space *X2,* if the feature embedding belongs to *L,* its label is 0. Otherwise, the label is 1. The binary dataset is utilized to train a shallow binary classification network *N2*, which contains several full connected layers. Then we remove the last classification layer of the network *N2* and extract the feature embedding of *X2,* which can be marked as *X3.* The size of feature space *X3* is 256-d.

Then, in the feature space *X3*, we use K-means to obtain the cluster centroid *C* of the feature embedding in *L*. Euclidean distance is adopted to measure the similarity between the cluster centroid *C* and each sample in *U*, which can be described by Eq (1):

$$D(i) = \|U(i) - C\|_2 \quad (1)$$

Where i is the index of the samples in *U, D*(*i*) describes the Euclidean distance between the i-th sample in *U* and the cluster centroid C. According to the sorted *D*, we pick out the top-K instances in *U* and label them, which should have the large distance with *C*. By doing this, we can choose

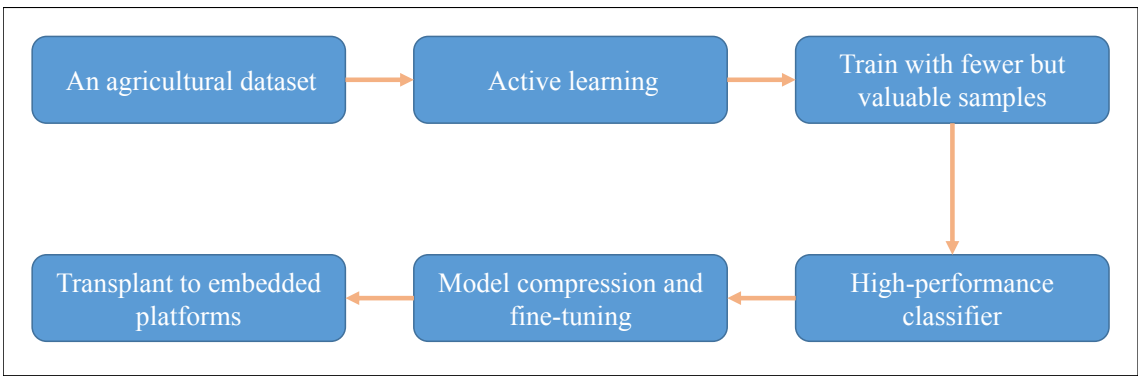


**Figure 3.** The overall framework.

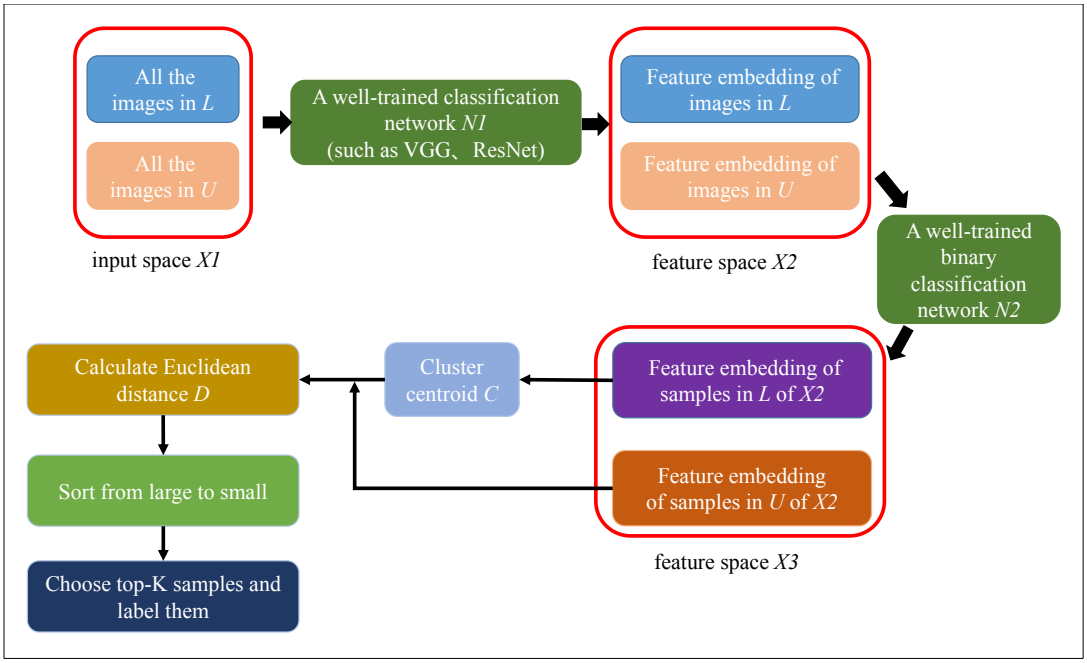


**Figure 4.** The proposed DBAL method.

the samples that are dissimilar to those in the labelled pool *L*, which may contain more useful information and cover the diversity of the whole dataset, finally help improve the classification accuracy.

### 2.4. Model compression method

With the wide application of deep learning, the performance of weed classification greatly improves, which benefits from the increase in the depth of the network. However, the large number of parameters results in an unaffordable computation burden for embedded devices with scant resources. Therefore, if we want to deploy the model on the embedded devices, compressing the model will be an effective method to accelerate the inference process of the network.

Figure 5 shows the procedure of model compression. Firstly, a large model should be pretrained, which may contain a huge number of parameters and redundant convolutional channels. Next, we choose the appropriate metric to measure the significance and prune the unimportant channels. Directly pruning sometimes hurt the performance, so fine-tuning is necessary to compensate the accuracy. Then we evaluate the performance of the pruned model and decide if it is practical to deploy on embedded devices. If not, we may need to iteratively prune.

It is vital to select the suitable metric to judge the importance. We take notice of BN layers, which is described as Eq (2):

$$y = \gamma \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta \tag{2}$$

Where x is the input of the BN layer, and y is the output. $\mu_B$ and $\sigma_B^2$ describe the mean and variance of the input in each mini-batch B. $\gamma$ and $\beta$ represent the scale and shift factor of the BN layer. BN layers keep the number of channels unchanged following convolutional layers. Each scale factor corresponds to a convolutional channel. From Eq (2), if the scale factor $\gamma$ is very small or close to zero, then the output y will perhaps be a constant and have less connection with the input. So, we use the scale factors of BN layers to measure the significance of convolutional channels, which will not introduce extra parameters.

To force the scale factors becoming sparse or close to zero, we punish them by L1 regularization. The modified loss function can be described by Eq (3):

$$L_{total} = loss_{CE} + \rho \sum_{\gamma \in T} |\gamma| \tag{3}$$

Where $loss_{CE}$ represents the cross-entropy loss function, which is widely used in the classification task. T is a set including all the scale factors. $\rho$ can balance the two items in $L_{total}$.

After training, we can obtain a model with sparse scale factors. Those convolutional channels corresponding to the small-scale factors will be pruned according to the setting threshold θ. The maximum threshold marked as $\theta_{max}$ is introduced to ensure that we will not remove all the channels in any layer. This means there will exist at least one channel in any layer after pruning. The threshold θ(< $\theta_{max}$) represents the pruning proportion we want. If the number of scale factors is M, then N (N = M*θ) channels

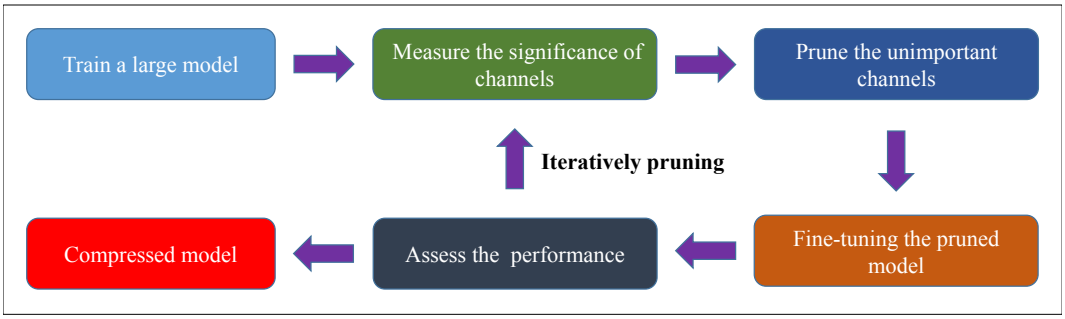


**Figure 5.** The procedure of model compression.

will be pruned. We remove the convolutional channels if their corresponding scale factors are more diminutive than the N-th value among the sorted scale factors. After pruning the accuracy may drop, so fine-tuning is used to compensate the accuracy. Then we evaluate the performance of the compressed model and determine whether transplanting to embedded devices is feasible. Otherwise, we will repeat the iterative procedure of pruning.

By model compression, the number of parameters and model size have degraded a lot. Then we transplant the compressed model to NVIDIA Jetson AGX Xavier. It has a 512-core volta GPU and 16GB memory. Enjoying the advantages of low power consumption and convenience, NVIDIA Jetson AGX Xavier has been widely used in many areas, such as agriculture, industry and so on. Our work will provide a valuable reference for the deployment of precision agriculture.

## 3. Results

### 3.1. Experiment setup for active learning

As for active learning, our code runs with one Titan Xp GPU in Centos system. The concrete development environment contains Python 3.6, CUDA 10.1, CUDNN 7.6.5, TensorFlow 1.14.0, Keras 2.3.1. We use VGG network architecture as the network *N1* for weed classification. All the images in the two different datasets are resized to 224 × 224. We split the two datasets to obtain training set and testing set by 8:2. The testing set is used for evaluating the model performance. The training set is divided into a small labelled pool *L* and a large unlabelled pool *U*. In the beginning, we set the initial size of *L* for 800. In active learning process, we use our proposed method and choose K (K = 600/800/1000) samples from *U*, which will be put into *L* and the number of images in *L* will be larger. If these samples are chosen for only one time, we cannot guarantee that they are the most dissimilar with *L*. So, we repeat the active learning process for 10 times, which means only K/10 samples are chosen for one time. Every time when K/10 samples are put into *L*, we retrain the binary classification network *N2* and continue the next step to choose the next K/10 samples. Only when K images are chosen and put into *L*, we retrain the VGG network *N1* using the images in *L* and continue.

### 3.2. Results for active learning

To verify the effectiveness, we compare the performance of our DBAL method with random sampling according to the test accuracy on the two datasets.

#### 3.2.1. Results on DeepWeeds dataset

Figure 6, Figure 7 and Figure 8 show the results for K = 600/800/1000 respectively of different methods on the DeepWeeds dataset. The baseline accuracy (91.5%) refers to using the whole training set of the DeepWeeds dataset to train the VGG model without active learning process and evaluate on the testing set. The x-axis, y-axis represent the current number of labelled images and test accuracy, respectively. We also calculate the ratio between the current number of labelled images and the total number of DeepWeeds dataset, such as 4.6% = 800/17,509.

From the results on the DeepWeeds dataset, when K = 600, our DBAL method can obtain the test accuracy of 86.88% with 21.7% amount of the dataset, 3% higher than random sampling, up to 94.95% of the baseline accuracy. As K = 800, our method can acquire the test accuracy of 90.75% with 32% amount of the dataset, 2.9% higher than random sampling, up to 99.18% of the baseline accuracy. When K = 1000, our method can reach the test accuracy of 89.25% with 27.4% amount of the dataset, 3.2% higher than random sampling, up to 97.54% of the baseline accuracy.

#### 3.2.2. Results on Grass-Broadleaf dataset

Figure 9, Figure 10 and Figure 11 show the results for K = 600/800/1000 respectively of different method on the Grass-Broadleaf dataset. The baseline accuracy (99.52%) means we use the whole training set of the Grass-Broadleaf dataset to train the VGG model without active learning process and evaluate on the testing set.

As shown in Figure 9, Figure 10 and Figure 11, for K = 600, our method can reach the test accuracy of 98.14% with 24.8% amount of the dataset, 1.4% higher than random

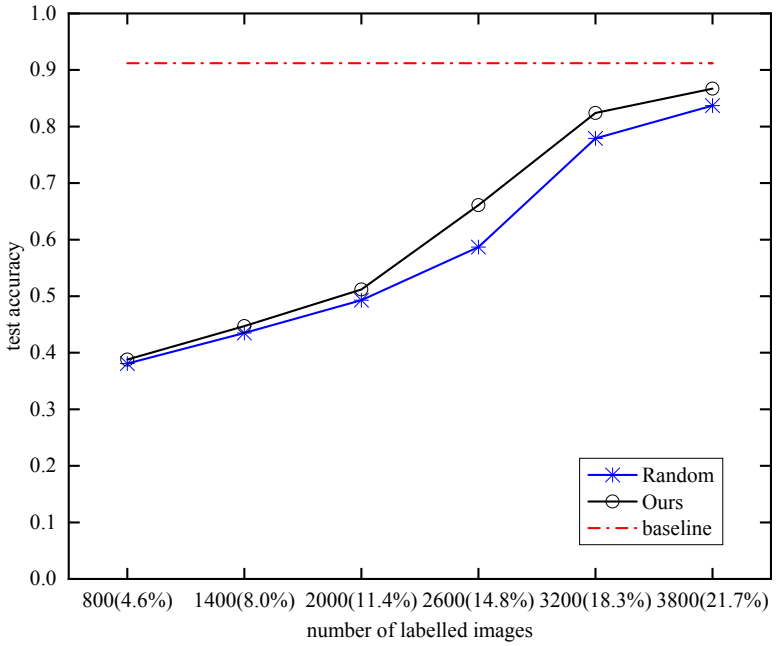


**Figure 6.** K = 600 on the DeepWeeds dataset.

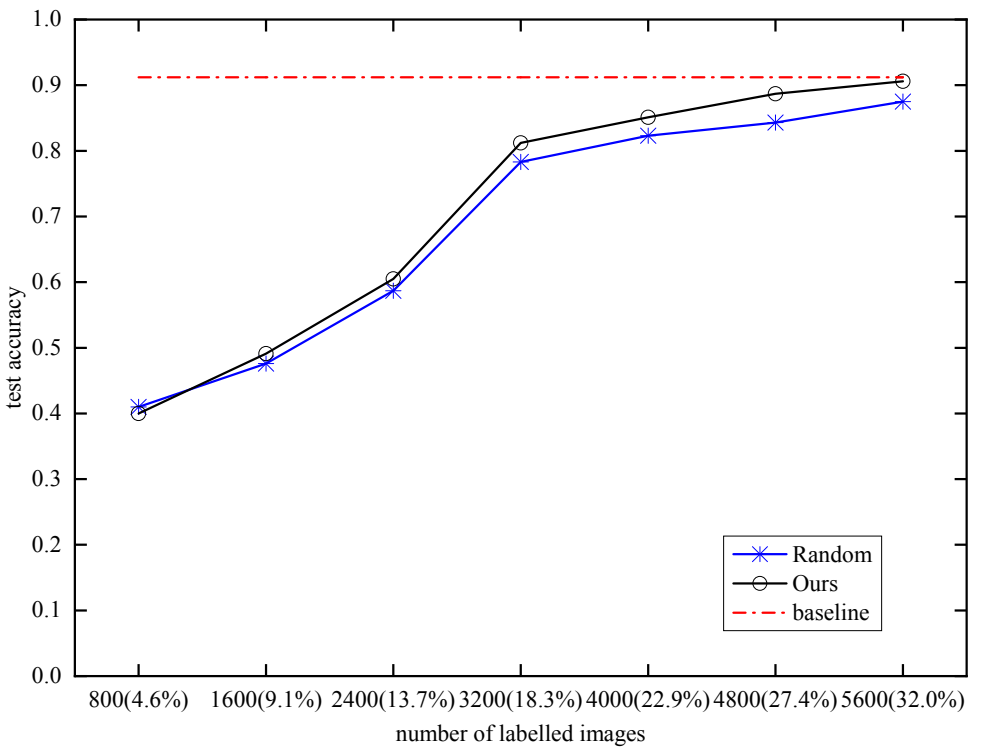


**Figure 7.** K = 800 on the DeepWeeds dataset.

sampling, up to 98.59% of the baseline accuracy. When K = 800, with 31.3% amount of the dataset, our method can obtain the test accuracy of 98.79%, 1.5% higher than random sampling, up to 99.2% of the baseline accuracy. As K = 1000, our method can get the test accuracy of 98.97% with 27.8% amount of the dataset, 2.2% higher than random sampling, up to 99.5% of the baseline accuracy.

### 3.3. Results for model compression

Model compression experiments are executed with VGG network on the DeepWeeds dataset. In detail, we set $\rho$ =

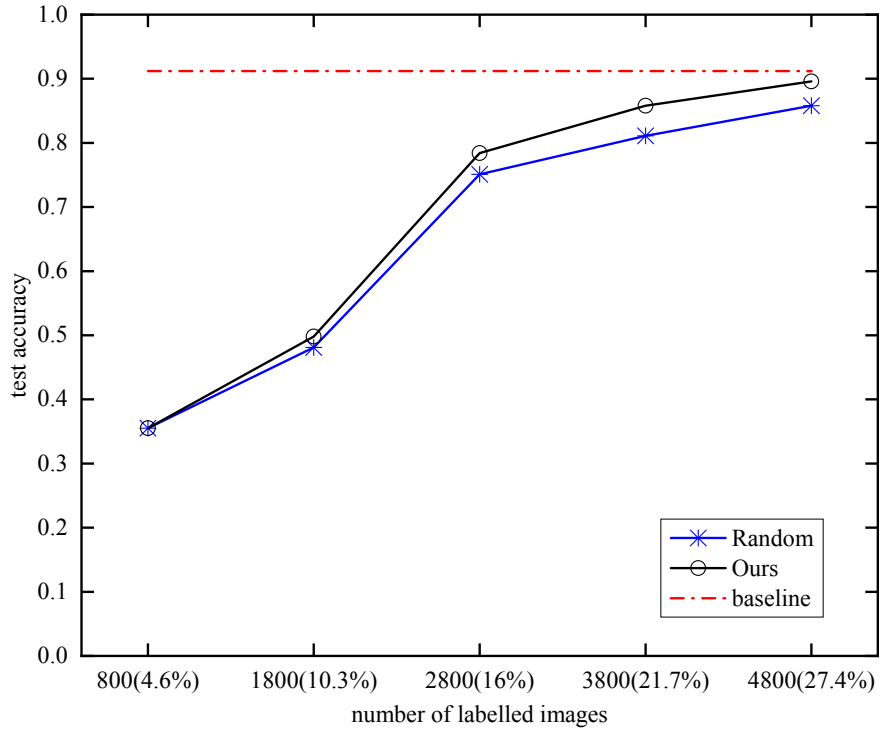


**Figure 8.** K = 1000 on the DeepWeeds dataset.

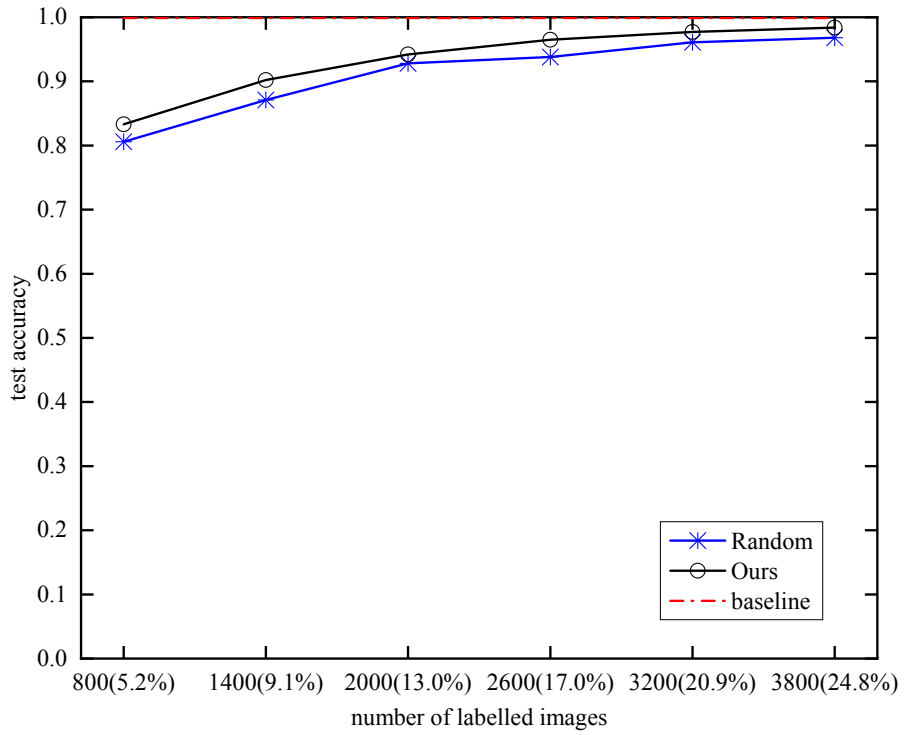


**Figure 9.** K = 600 on the Grass-Broadleaf dataset.

0.0001 to balance the Eq (3) and θ = 0.3/0.5/0.7. Different values of θ represent different channel pruning ratios. We choose the model training with 5600 labelled images, 32% amount of DeepWeeds dataset, under the condition of K =

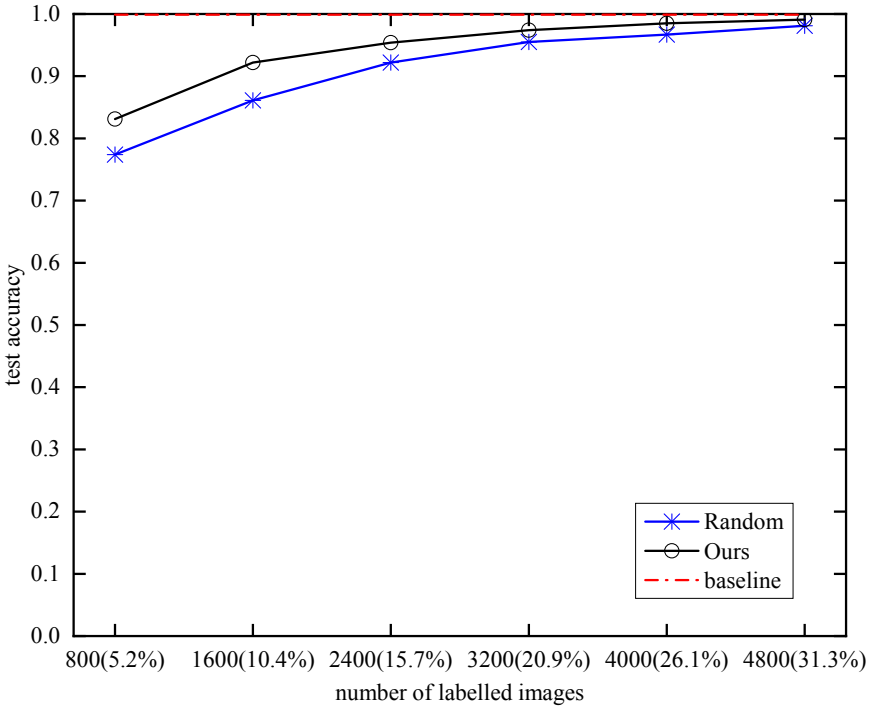


**Figure 10.** K = 800 on the Grass-Broadleaf dataset.

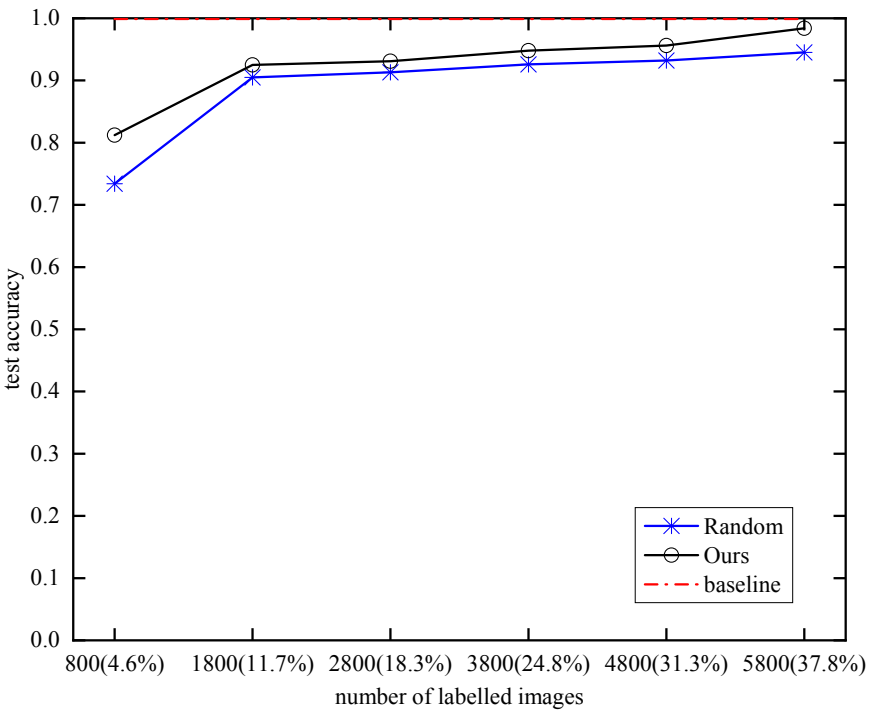


**Figure 11.** K = 1000 on the Grass-Broadleaf dataset.

800 in Figure 8, and execute the compression process. The baseline test accuracy (90.75%) means we directly evaluate the model performance on the testing set.

Table 3 shows the results of model compression. Directly pruning will cause the drop of performance. So, the test accuracy under different conditions of θ in Table

**Table 3.** Results for model compression.

| | Test accuracy | Parameters | Model size | Compression ratio |
|---|---|---|---|---|
| baseline | 90.75% | 14.7M | 117.9MB | - |
| θ = 0.3 | 88.88% | 6.1M | 49.2MB | 58.3% |
| θ = 0.5 | 89.53% | 2.8M | 22.2MB | 81.2% |
| θ = 0.7 | 89.75% | 1.1M | 8.6MB | 92.7% |

3 is calculated after sparsity training, pruning and fine-tuning process. We also calculate the compression ratio for model size. The VGG model can be compressed from 117.9 MB to 8.6 MB, corresponding to a compression ratio of 92.7%, while without remarkable degradation of accuracy, about 1%. Finally, the compressed model is transplanted to NVIDIA Jetson AGX Xavier, running at 192 fps, which will beneficial to farmers for weed identification.

## 4. Discussion

When it comes to intelligent agriculture with CNNs (Li and Chao, 2020; Li and Chao, 2021), there are two main problems. First, deep learning based CNNs need a large amount of data. Obviously labelling these datasets is difficult and costs much of the time. Second, with the increasing depth of the network, the performance becomes better and better. However, there are more and more parameters in the model. Thus, to solve the above two problems, we propose a new active learning method DBAL to choose valuable samples from the unlabeled pool, using fewer samples to cover the diversity of the whole dataset, which is related with the current few-shot learning (Yang et al, 2022), and then we remove the unimportant channels of the network to reduce the parameters and model size. By calculating the distance between each sample in unlabeled pool and the cluster centroid of labelled pool, our method DBAL can choose samples that are dissimilar to those in the labelled pool, which can contain more valuable information. In other words, our method can use fewer samples to cover the diversity of the whole dataset as much as possible.

With the rapid improvement of Internet of Things, deep learning models have made remarkable progress in intelligent agriculture. However, we cannot directly apply those high-end servers, which is obviously impractical. Low power consumption is a main trend. So, model compression is executed to reduce the number of parameters and model size. We apply L1 regularization on the scale factors of BN layers to find those unimportant channels and prune. Experiment results show that we can reach a compression ratio of 92.7% with tolerable 1% accuracy drop. The model size after compression is only 8.6MB, which will occupy less running memory. This means there are many unimportant channels in the CNNs (Liu et al, 2020). We can prune those according to our demand without obvious accuracy drop. Then we deploy the compressed model to NVIDIA Jetson AGX Xavier, with the running speed of 192fps, which contributes to the practical smart applications. In the future, this work can lay a foundation for other research works about few-shot learning, data quality assessment, and model deployment in agriculture.

## 5. Conclusion

Considering the two problems of large amount of labelled data and numerous model parameters of deep CNNs, in this paper, we develop a framework containing active learning and model compression. Concretely, we propose a DBAL method to choose a few yet valuable samples to cover the diversity of the whole dataset according to metric.

We validate our method on two different datasets for weed classification and the results demonstrate that our DBAL method is competitive. Our method can achieve 90.75% test accuracy with 32% amount of the dataset, up to 99.18% of the baseline accuracy.

Then model compression is executed to reduce the number of parameters and model size. With a compression ratio of 92.7%, the model is compressed from 117.9 MB to 8.6 MB, with 1% accuracy drop. The compressed model runs on embedded NVIDIA Jetson AGX Xavier with the speed of 192 fps. This work can provide some reference for the following research based on limited samples and model deployment.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 32101612).

## References


Gu L, He F, Yang S (2019). Crop classification based on deep learning in northeast China using sar and optical imagery. In: 2019 SAR in Big Data Era (BIGSARDATA); pp. 1-4. doi: 10.1109/BIGSARDATA.2019.8858437

He Y, Liu P, Wang Z, Hu Z, Yang Y (2019). Filter pruning via geometric median for deep convolutional neural networks acceleration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; pp. 4340-4349. doi: 10.48550/arXiv.1811.00250

Jiang CK, Liu ZL, Li XY, Ercisli S, Ma JQ et al. (2021). Non-volatile metabolic profiling and regulatory network analysis in fresh shoots of tea plant and its wild relatives. Frontiers in Plant Science 12. doi: 10.3389/fpls.2021.746972

Kozłowski M, Górecki P, Szczypiński PM (2019). Varietal classification of barley by convolutional neural networks. Biosystems Engineering 184: 155-165. doi: 10.1016/j.biosystemseng.2019.06.012

Li Y, Cao Z, Lu H, Xu W (2020). Unsupervised domain adaptation for in-field cotton boll status identification. Computers and Electronics in Agriculture 178: 105745. doi: 10.1016/j.compag.2020.105745

Li Y, Chao X (2020). ANN-based continual classification in agriculture. Agriculture 10 (5): 178. doi: 10.3390/agriculture10050178

Li Y, Chao X (2021). Distance-entropy: an effective indicator for selecting informative data. Frontiers in Plant Science 12. doi: 10.3389/fpls.2021.818895

Li Y, Chao X (2021). Semi-supervised few-shot learning approach for plant diseases recognition. Plant Methods 17 (1): 1-10.

Li Y, Chao X (2021). Toward sustainability: trade-off between data quality and quantity in crop pest recognition. Frontiers in plant science 12. doi: 10.3389/fpls.2021.811241

Li Y, Chao X, Ercisli S (2022). Disturbed-entropy: A simple data quality assessment approach. ICT Express. doi: 10.1016/j.icte.2022.01.006

Li Y, Gu S, Gool LV, Timofte R (2019). Learning filter basis for convolutional neural network compression. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; pp. 5623-5632. doi: 10.1109/ICCV.2019.00572

Li Y, Nie J, Chao X (2020). Do we really need deep CNN for plant diseases identification? Computers and Electronics in Agriculture 178: 105803. doi: 10.1016/j.compag.2020.105803

Li Y, Yang J (2020). Few-shot cotton pest recognition and terminal realization. Computers and Electronics in Agriculture 169: 105240. doi: 10.1016/j.compag.2020.105240

Li Y, Yang J (2021). Meta-learning baselines and database for few-shot classification in agriculture. Computers and Electronics in Agriculture 182: 106055. doi: 10.1016/j.compag.2021.106055

Li Y, Yang J, Wen J (2021). Entropy-based redundancy analysis and information screening. Digital Communications and Networks. doi: 10.1016/j.dcan.2021.12.001

Lin M, Ji R, Wang Y, Zhang Y, Zhang B et al. (2020). Hrank: Filter pruning using high-rank feature map. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; pp. 1529-1538. doi: 10.1109/CVPR42600.2020.00160

Liu N, Ma X, Xu Z, Wang Y, Tang J et al. (2020). Autocompress: An automatic dnn structured pruning framework for ultra-high compression rates. In: Proceedings of the AAAI Conference on Artificial Intelligence; pp. 4876-4883. doi: 10.1609/aaai.v34i04.5924

Liu Z, Li J, Shen Z, Huang G, Yan S et al. (2017). Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE international conference on computer vision; pp. 2736-2744. doi: 10.48550/arXiv.1708.06519

Lu Y, Young S (2020). A survey of public datasets for computer vision tasks in precision agriculture. Computers and Electronics in Agriculture 178: 105760. doi: 10.1016/j.compag.2020.105760

Mohanty SP, Hughes DP, Salathé M (2016). Using deep learning for image-based plant disease detection. Frontiers in plant science 7: 1419. doi: 10.3389/fpls.2016.01419

Nie J, Li Y, She S, Chao X (2019). Magnetic shielding analysis for arrayed Eddy current testing. Journal of Magnetics 24 (2): 328-332. doi: 10.4283/JMAG.2019.24.2.328

Olsen A, Konovalov DA, Philippa B, Ridd P, Wood JC et al. (2019). DeepWeeds: A multiclass weed species image dataset for deep learning. Scientific reports 9 (1): 1-12. doi: 10.1038/s41598-018-38343-3

Rahman CR, Arko PS, Ali ME, Khan MAI, Apon SH et al. (2020). Identification and recognition of rice diseases and pests using convolutional neural networks. Biosystems Engineering 194: 112-120. doi: 10.1016/j.biosystemseng.2020.03.020

Rastegari M, Ordonez V, Redmon J, Farhadi A (2016). Xnor-net: Imagenet classification using binary convolutional neural networks. In: European conference on computer vision; pp. 525-542. doi: 10.48550/arXiv.1603.05279

Villa-Henriksen A, Edwards GT, Pesonen LA, Green O, Sørensen CAG (2020). Internet of Things in arable farming: Implementation, applications, challenges and potential. Biosystems engineering 191: 60-84. doi: 10.1016/j.biosystemseng.2019.12.013

Yan Z, Xing P, Wang Y, Tian Y (2020). Prune it yourself: Automated pruning by multiple level sensitivity. In: 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR); pp. 73-78. doi: 10.1109/MIPR49039.2020.00022

Yang J, Guo X, Li Y, Marinello F, Ercisli S et al. (2022). A survey of few-shot learning in smart agriculture: developments, applications, and challenges. Plant Methods 18 (1): 1-12. doi: 10.1186/s13007-022-00866-2

Yang S, Gu L, Li X, Jiang T, Ren R (2020). Crop classification method based on optimal feature selection and hybrid CNN-RF networks for multi-temporal remote sensing imagery. Remote Sensing 12 (19): 3119. doi: 10.3390/rs12193119

Yin C, Qian B, Cao S, Li X, Wei J et al. (2017). Deep similarity-based batch mode active learning with exploration-exploitation. In: 2017 IEEE International Conference on Data Mining (ICDM); pp. 575-584. doi: 10.1109/ICDM.2017.67

Yoo D, Kweon IS (2019). Learning loss for active learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; pp. 93-102. doi: 10.1109/CVPR.2019.00018

Zhong Y, Hu X, Luo C, Wang X, Zhao J et al. (2020). WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF. Remote Sensing of Environment 250: 112012. doi: 10.1016/j.rse.2020.112012