



OPEN ACCESS

EDITED BY

Zenghui Zhang,
Shanghai Jiao Tong University, China

REVIEWED BY

Weitao Chen,
China University of Geosciences Wuhan,
China
Weiwei Guo,
Tongji University, China

*CORRESPONDENCE

Beibei Zhou

✉ happyangle222@gmail.com

Dapeng Ye

✉ ydp@fafu.edu.cn

†These authors have contributed equally to this work

RECEIVED 13 June 2024

ACCEPTED 16 October 2024

PUBLISHED 13 February 2025

CITATION

Chen Y, Zhou B, Xiaopeng C, Ma C, Cui L, Lei F, Han X, Chen L, Wu S and Ye D (2025) A method of deep network auto-training based on the MTPI auto-transfer learning and a reinforcement learning algorithm for vegetation detection in a dry thermal valley environment. *Front. Plant Sci.* 15:1448669. doi: 10.3389/fpls.2024.1448669

COPYRIGHT

© 2025 Chen, Zhou, Xiaopeng, Ma, Cui, Lei, Han, Chen, Wu and Ye. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

A method of deep network auto-training based on the MTPI auto-transfer learning and a reinforcement learning algorithm for vegetation detection in a dry thermal valley environment

Yayong Chen^{1,2†}, Beibei Zhou^{1,2*†}, Chen Xiaopeng^{1,2}, Changkun Ma^{1,2}, Lei Cui³, Feng Lei⁴, Xiaojie Han⁵, Linjie Chen^{6,7}, Shanshan Wu^{1,2} and Dapeng Ye^{7,8*}

¹State Key Laboratory of Eco-hydraulics in Northwest Arid Region, Xi'an University of Technology, Xi'an, China, ²School of water resources and hydro-electric engineering of XUT, Xi'an University of Technology, Xi'an, China, ³China Renewable Energy Engineering Institute, Beijing, China, ⁴Central South Survey and Design Institute Group Co., Ltd., Changsha, China, ⁵China Electric Construction Group Beijing Survey and Design Institute Co., Beijing, China, ⁶Center for Artificial Intelligence in Agriculture, School of Future Technology, Fujian Agriculture and Forestry University, Fuzhou, China, ⁷Fujian Key Laboratory of Agricultural Information Sensing Technology, Fujian Agriculture and Forestry University, Fuzhou, China, ⁸College of Mechanical and Electrical Engineering, Fujian Agriculture and Forestry University, Fuzhou, China

UAV image acquisition and deep learning techniques have been widely used in field hydrological monitoring to meet the increasing data volume demand and refined quality. However, manual parameter training requires trial-and-error costs (T&E), and existing auto-trainings adapt to simple datasets and network structures, which is low practicality in unstructured environments, e.g., dry thermal valley environment (DTV). Therefore, this research combined a transfer learning (MTPI, maximum transfer potential index method) and an RL (the MTSA reinforcement learning, Multi-Thompson Sampling Algorithm) in dataset auto-augmentation and networks auto-training to reduce human experience and T&E. Firstly, to maximize the iteration speed and minimize the dataset consumption, the best iteration conditions (MTPI conditions) were derived with the improved MTPI method, which shows that subsequent iterations required only 2.30% dataset and 6.31% time cost. Then, the MTSA was improved under MTPI conditions (MTSA-MTPI) to auto-augmented datasets, and the results showed a 16.0% improvement in accuracy (human error) and a 20.9% reduction in standard error (T&E cost). Finally, the MTPI-MTSA was used for four networks auto-training (e.g., FCN, Seg-Net, U-Net, and Seg-Res-Net 50) and showed that the best Seg-Res-Net 50 gained 95.2% WPA (accuracy) and 90.9% WIoU. This study provided an effective auto-training method for complex vegetation information collection, which provides a reference for reducing the manual intervention of deep learning.

KEYWORDS

vegetation detection, segmentation deep learning, network training automatic, data augmentation automatic, reinforcement learning for DL, auto-DL method

Highlights

- The MTPI method was improved to auto gain best transfer learning conditions.
- A reinforcement learning algorithm under MTPI (MTPI-MTSA) was proposed for dataset auto-augmented.
- Four deep vegetation detection networks were auto-trained with the MTPI-MTSA.

1 Introduction

Vegetation detection with DL (deep learning) in field UAVS [Unmanned Aerial Vehicle System(s)] images is an important fundamental technology, which has been widely used in regional hydrological information monitoring (Ang and Seng, 2021; Yang et al., 2022). Although (DL) networks, simulating animal neural structures [e.g., human or mammalian visual systems (Hassabis et al., 2017)], can accurately extract image features for (semantic) segmentation after training iterations, the training process may require lots of human experience and T&E cost (Li et al., 2020). In particular, those vegetation detection networks in unstructured environments [e.g., DTV (dry thermal valley)], which have numerous network layers, complex structures, convolutional kernels, and weighting/bias values (Saleem et al., 2021; Gentilhomme et al., 2023). Therefore, many scholars have been focusing on the DL practicality and effectiveness, such as network structure improvement (Lin et al., 2023; Liu et al., 2023; Towfek and Khodadadi, 2023), data improvement (Salcedo-Sanz et al., 2020; Kamarudin et al., 2021; Li et al., 2021), and operation methods (Al-Hyari and Abu-Faraj, 2022).

The network structure determines the functional implementation and affects the overall feature extraction ability, generalization ability, computational efficiency, and parameter number (Niu et al., 2021). So, network optimization enhances the network accuracy, improves the training effect, reduces the training iterations, and reduces the labor experience and T&E costs in training process (Shrestha and Mahmood, 2019). For example, Ouyang, S. (Ouyang et al., 2024) developed an LSBP-net structure, which incorporated a U-Net structure to reduce the vegetation influence on the lithological spectral characteristics of optical remote sensing, and the result showed a 13.94% accuracy improvement. Zhao, S.Y. (Zhao et al., 2021), Wang, P. (Wang et al., 2021) and Yang, L. (Yang et al., 2023) pointed out that adding an attention mechanism, which mimics a human vision or cognitive focus, can enhance network performance. The attention mechanisms might enhance the network performance by increasing the task-related information weight through selective reinforcement mechanisms (Ni et al., 2023). In addition, LSTM (Long Short-Term Memory) structure that mimics biological memory (Xia et al., 2020; Chen X. et al., 2023), PPM (Pyramid Pooling Module) structure (Lu et al., 2021; Pu et al., 2022), and AE (autoencoder or Self-Encoder) structure (Yang et al., 2019; Yu et al., 2023) are also common methods, when they reduce the gradient vanishing problem and improve the training effect by improving the patterns and information transmission pathways (Hu et al., 2022).

However, optimizing the network structure still requires a great deal of human experience and T&E (Xu et al., 2023) in complex field environmental dataset conditions.

On the other hand, the data improvement can increase the network training rate, thus it can increase the validity of the dataset to improve the network accuracy (Nikolados et al., 2022) and reduce the time consumption (García et al., 2016). Firstly, a higher validity dataset can make it easier for the network to learn the desired features, and secondly, a valid dataset can reduce the dataset size (Ali et al., 2021), which reduces the DL training time (Rosende et al., 2023). However, when in unstructured environments, many researchers focusing on multiple data sources (Gao et al., 2020) in nowadays. Including the multi-scale same data type (García et al., 2016), multi-source data in the same form (Ding et al., 2022), and multiple data (Wang Y. et al., 2020). For example, Wei, D. P. (Weit et al., 2022) and Marzougui, A. (Marzougui et al., 2023) used multi-scale images to detect the vegetation health status, and the fusion scale method showed better performance than the traditional detection. Similarly, DL is available for fusion of different data sources, e.g., Mu, C. H. (Mu et al., 2020) used fused information from imagery and hyperspectral data, Maimaitijiang, M. (Fritsch, 2020) used fused satellite and UAV data, and Kang and Wang (Kang and Wang, 2023) used fusion multispectral and SAR (Synthetic Aperture Radar) data. In addition, DL for fusing multimodal methods is also rapidly developing field in recent years, e.g., Patil, R. R. (Patil and Kumar, 2022) proposed a CNN network fusing meteorological and image data, Nasir, I. M. (Nasir et al., 2021) proposed a VGG 19 fusing temperature, wind speed acquired by IoT (Internet of Things), and image information, and Zheng, W. Q. (Zheng et al., 2022) proposed a Point-Net with RGB (Red/Green/Blue) images and LiDAR (Light Detection and Ranging) data. Although data fusion methods complement the network features in different dimensions to improve their performance, the data accumulation process and the data validity verification process require a lot of T&E cost and human experience.

The DL automated method implementation represents an effective approach to reducing T&E cost, and human experience. For example, the auto-PyTorch framework introduced by Zimmer, L. (Zimmer et al., 2021), the cloud auto-ML (automated Machine Learning) platform introduced by Santu, S. K. (Santu et al., 2021), and the auto-CVE system introduced by David, R. (Reyer, 2022) have been identified as effective development platforms that shorten training time, and reduced T&E. But, the simplicity necessity and performance requirements of platform constrain the DL flexibility and efficacy in complex vegetation environments. In addition, auto-DL based on intelligent algorithms could be an effective approach with minimal human intervention and T&E costs, as exemplified by the genetic algorithm proposed by Srivastava, A. (Srivastava et al., 2021), and Xiao, X. L. (Xiao et al., 2020). Furthermore, the swarm algorithm (Yeh et al., 2021), auto-stopping genetic algorithm (Montes et al., 2023), and evolutionary algorithm (Al-Hyari and Abu-Faraj, 2022) can be employed to reduce the necessity for human intervention and T&E. However, these existing auto methods involve relatively simple network structures and datasets (need to improve the practicality of those auto methods), for instance, Sun, Y. (Sun et al., 2020) designed an auto method for 10-layer networks. In

conclusion, his study has proposed an automated training process for dataset augmentation and deep vegetation detection networks using transfer learning (Chen Y. et al., 2023) and reinforcement learning algorithms. Even in the complex vegetation of DTV, the proposed auto-training method requires no human intervention, ensuring network versatility and practicality.

2 Materials

2.1 Data collection and dataset

Data collection was carried out in June 2021 near Qiaojia County, Yunnan Province, China (east 102° 53' 11", north 11° 9' 10", as Figure 1A), on a sunny, windless day, with steady humidity and barometric pressure. A small consumer-level 4-rotor UAVS, Phantom 4 v2 (DJI Innovation Technology Co., Shenzhen, China, as shown in Figure 1B), was used for the data collection. The CMOS (Complementary Metal Oxide Semiconductor) sensor of the UAVS camera is 4800×6400 p² (pixels×pixels), the lens is 3.5 mm and the image resolution is 43.4 mm². During image acquisition, the UAV flight altitude was 200 m, and the control mode was automatic control + manual model. The data collection consisted of 300 images with random occurrences in each image including mountains, buildings, water, trees, shrubs, grass, and sand.

During data collection, the UAV was operated by path planning and automatic flight control, and the remote-control system was an Android smartphone (HUAWEI P20 PRO, Huawei, Shenzhen, China) + DJI Pilot APP. The remote control was equipped with an "ON/OFF" switch, which could be switched to a manual remote controller + human judgment in the case of emergencies (encountering bird flying objects or high-altitude obstacles). Table 1 shows the parameters set by the UAV system during data acquisition, including positioning accuracy, UAV parameters, and camera parameters (RGB camera).

All the 300 UAV-acquired field images were split into 30,000 subregions of size 250×250 p², noted as im_i , ($i = 1, 2, \dots, M$); $M =$

30,000. The image content consists of various vegetated and non-vegetated elements including four categories, TR (Tree Region), SR (Shrub Region), GR (Grass Region), and NVR (Non-Vegetation Region), where the NVR may appear as buildings, water, roads, rocky, sandy, etc. The pixel regions of the dataset samples were manually labeled to distinguish different classes, such as TR (blue, (0,0,255), meaning RGB= (0,0,255), RGB means Red/Green/Blue channel values), SR [green, (0,255,0)], GR [red, (255,0,0)], and NVR [magenta, (255,0,255)], notated as lab_i , ($i = 1, 2, \dots, M$). All the obtained image examples im_i and labels mapping lab_i were correspondingly combined into a dataset (D , $D = \{im_i, lab_i\}$), and Figure 2 shows several random samples of im_i in D . The data in D was randomly divided into three parts (rounding up to the nearest integer) by 6:2:2 for networks training, validation, and testing, denoted as the training set (TD), validating set (VD), and testing set (SD).

2.2 The four deep networks and five evaluating parameters

Figure 3 gives the four common deep network structures used in this study, including Seg-Net (He et al., 2015; Badrinarayanan et al., 2017), FCN (Long et al., 2015), U-Net (Ronneberger et al., 2015), and Seg-Res-Net 50 (He et al., 2016; Chen et al., 2018) (the resnet 50 for segmentation). Among them, FCN is a (semantic) segmentation network with a full convolutional structure proposed by Long, J., which is fast and has less memory consumption. The Seg-Net is a convolutional neural network with encoder-decoder structures, which provides high resolution and detail preservation capability when processing large-scale images or increasing the layer depth. The U-Net is a network structure with U-shape down-sampling and up-sampling, which is simple and effective. The Seg-Res-Net 50 (Chen et al., 2018) is a high accuracy and few parameters segmentation network using residual blocks, obtained by further modification of the classified Res-Net 50 (Mandal et al., 2021). Compared with the Res-Net 50, it contains additional components

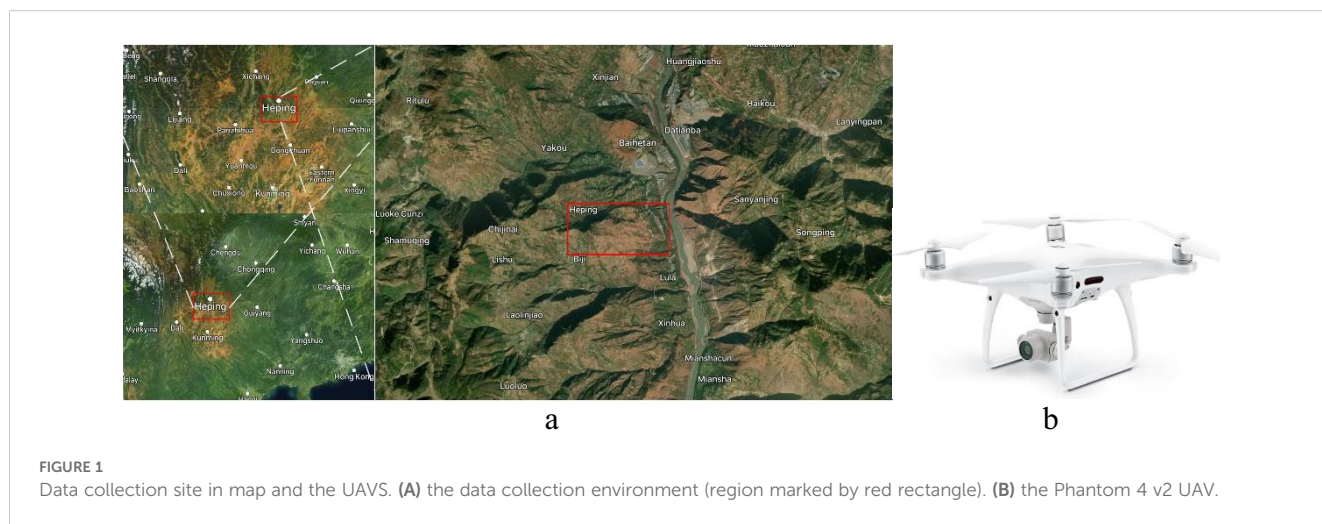


TABLE 1 The parameters of UAVS for data acquisition.

UAVS	Phantom 4 v2
Lens FOV	84
Equivalent focal length	35 mm
Positioning mode	GPS/GLONASS
Vertical positioning accuracy	± 0.1 m
Horizontal positioning accuracy	± 0.3 m
Image acquisition mode	no hover shooting
Resolution of location	Less than 50 mm
Flight altitude	<500 m
Flight speed	6 m/s
Aperture	f/2.8–f/11
Camera model	FCS400
Shutter	1/2000
Resolution	4800 × 6400
ISO range	100–12,800
Photo format	JPEG/DNG*

*Only JPEG was used in this study.

used to implement pixel processing capabilities, such as the inverse convolution part and the jump-joining part in Figure 3D.

In this study, the runtime environment used to construct, train, optimize and evaluate the network was MATLAB 2023b, running on a PC with Win10, equipped with a 24GB NVIDIA GeForce RTX 4090 GPU (graphics card) (12GHz), a 12th generation Intel (R) Core

(TM) i5-12600KF CPU (3.70 GHz) and 64GB DDR4 RAM (3600MHz).

To evaluate the performance of the four networks, five parameters were evaluated in this study, including WPA (Weighted Pixel Accuracy), WPP (Weighted Pixel Precision), WPR (Weighted Pixel Recall), WPF1 (Weighted Pixel F1 Score), and WIoU (Weighted Intersection over Union). Among them, WPA characterizes the totally correctness effect of pixel detection, as shown in (Equation 1).

$$WPA = \sum_{i=1}^C (p_i \times acc_i) \\ = 1/C \sum_{i=1}^C p_i \times (TP_i + TN_i) / (TP_i + FP_i + TN_i + FN_i) \quad (1)$$

Where, i is the type of vegetation and $i = 1, 2, \dots, C$ ($C=4$). TP_i is the number of true positive pixels, TN_i is the number of true negative pixels, FP_i is the number of false positive pixels, and FN_i is the number of false negative pixels. p_i is the probability that the i -th planted pixel accounts for the overall dataset images as shown in (Equation 2).

$$p_i = \sum_{c=1}^M |\{C_c\}|_i / \sum_{j=1}^C \sum_{i=1}^M |\{C_c\}|_i \quad (2)$$

when $|\{C_c\}|_i$ means the total number of pixel eligible for the i -th image in the dataset (e.g., dataset D). j means the j -th vegetation types ($C=4$). The WPP characterizes the correctness effect of pixel detection, as shown in (Equation 3).

$$WPP = \sum_{i=1}^C (p_i \times precision_i) \\ = 1/C \sum_{i=1}^C p_i \times (TP_i) / (TP_i + FP_i) \quad (3)$$

The WPR characterizes the ability of networks to find all positive pixels, as (Equation 4).

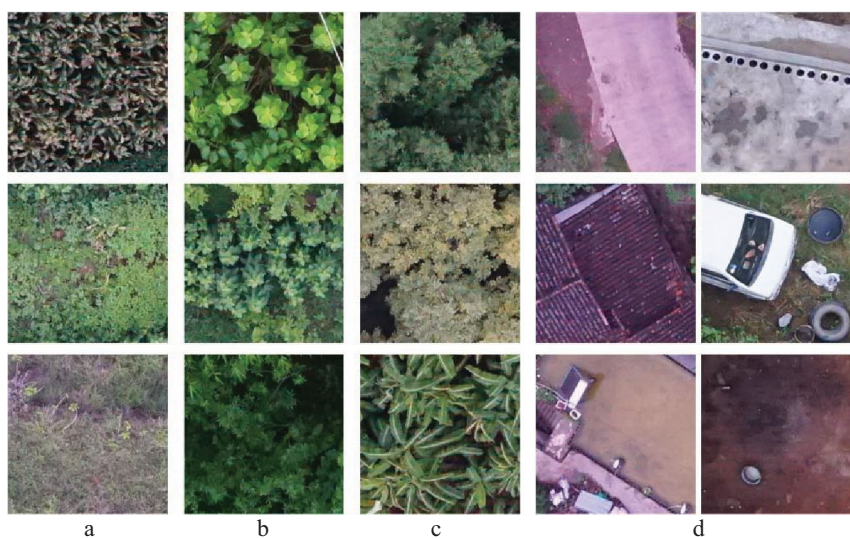


FIGURE 2 Several sample images. (A) random samples of pure GR. (B) pure SR. (C) pure TR. (D) pure NVR.

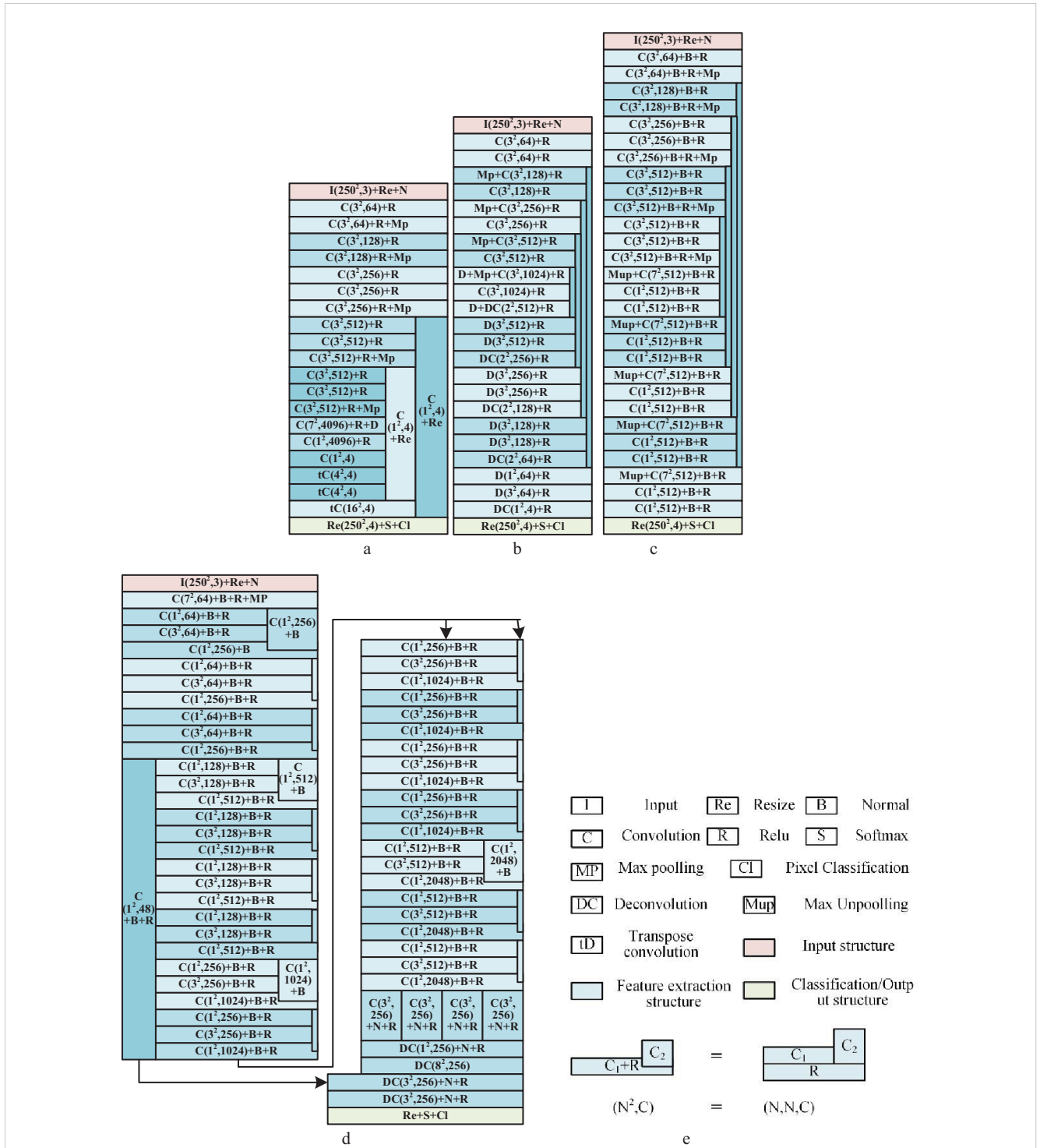


FIGURE 3 The four deep networks for vegetation detection in DTW. **(A)** The structure of FCN. **(B)** The structure of Seg-Net. **(C)** The structure of U-Net. **(D)** The structure of Seg-Res-Net 50. **(E)** Legends of networks. The size of each network layer in networks was labeled as (N^2, C) or (N, N, C) , where the (N^2, C) means the layer filter size equivalent to $N \times N \times C$. The C is the convolution layer, DC means the deconvolution layer, MP denotes the max pooling layer, B denotes the batch normalization layer, S is the SoftMax layer, R is the relu activation layer, and Re means the resize or scaling layer.

$$WPR = \sum_{i=1}^C (p_i \times recall_i) = \sum_{i=1}^C p_i \times TP_i / (TP_i + TN_i) \quad (4)$$

And the WPF1 is the harmonic mean of precision and recall ($0 \leq WF1 \leq 1$), with higher values showing better method

performance. When the precision and recall of the method are both high, the WF1 achieves a maximum value of 1. Therefore, the WF1 is a metric that combines the precision and recall of networks and can be used to evaluate the overall network performance, expressed as (Equation 5).

$$WPF1 = 2 \sum_{i=1}^C p_i \times (\text{precision}_i \times \text{recall}_i) / (\text{precision}_i + \text{recall}_i) \quad (5)$$

The $WIoU$ is the intersection over union metric, which measures the overlap between the predicted and ground truth masks, expressed as (Equation 6).

$$WIoU = \sum_{i=1}^C p_i \times TP_i / (TP_i + FP_i + FN_i) \quad (6)$$

3 Methods

The Figure 4 shows the main auto-training method process presented in this manuscript consists of three main parts, including the MTPI conditions derivation, auto dataset augmentation with MTPI-MTSA (multi-TSA based on the MTPI method), and auto network training with MTPI-MTSA.

3.1 Optimization of the MTPI transfer learning for deep networks auto-training

Since DL training is time-consuming and re-training of automatic methods multiplies the time consumption, it is necessary to reduce the training time to ensure automation usefulness. So, the MTPI (Chen Y. et al., 2023) (Maximum Transfer Potential Index method), which was proposed in the previous research work was improved and used in this study, to minimize the manual experience bias in complex field environments. The MTPI implements a comprehensive evaluation of the potential for a network training (TPI, Transfer Potential Index) index in both accuracy and time consumption. The minimum dataset and iteration times would be determined by obtaining the maximum TPI corresponding to the transfer training condition under low overfitting risk.

The MTPI conditions [dataset and time, or TDS and LT (learning time)] determining was obtained by maximizing the quotient of the TPI_{acc} (Accuracy Transferable Index) and the TPI_{time} (Time Transferable Index), as shown in (Equation 7).

$$TPI = TPI_{acc} / TPI_{time} \quad (7)$$

In this manuscript, the MTPI has been improved so that the potential indices expressed in TPI_{acc} and TPI_{time} refer to the actual results of the pre-training process, whereas the original MTPI ($MTPI_0$) was used to derive the TPI indices for each network layer. To obtain the MTPI conditions, $TDS = 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1$ ($\times 100\%$) were set and the TPI_{acc} and TPI_{time} corresponding to TDS were derived. The relationship between TDS and TPI_{acc} was solved by logarithmic fitting to obtain $TPIC_{acc}$ (TPI_{acc} curve), and the correlation between TDS and TPI_{time} was exponentially fitted to obtain $TPIC_{time}$ (TPI_{time} curve). Thereafter, the $TPIC$ (TPI curve) was obtained from TDS in the range of 0 to 1 by using (Equation 7) and the corresponding training conditions (including TDS and LT) for the maximum value of $TPIC$ (MTPI) were obtained for the subsequent automatic training process based on the MTPI.

In this study, only the vegetation segmentation problem in the DTV environment was been discussed, so the structural complexity

and task complexity of the network would be constant. Therefore, it can be assumed that the size of LT is related to TDS and TPI_{acc} is expressed as (Equation 8).

$$TPI_{acc} = WPA(TDS) \quad (8)$$

Where TDS is the migrated dataset size and WPA is the previously mentioned accuracy obtained through (Equation 3). In addition, TPI_{time} is expressed as the relative number of iterations corresponding to TPI_{acc} .

3.2 The MTPI-based reinforcement learning for dataset auto-augmentation

Data augmentation techniques (Shorten and Khoshgoftaar, 2019) is a common methods to improve the data effectiveness and reduce the overfitting risk for network training, such as Abayomi-Alli, O.O. (Abayomi-Alli et al., 2021), Ottoni, A.L.C. (Ottoni et al., 2023), and Ayhan, B. (Ayhan et al., 2020) pointed out. However, the processes and range parameters they used in data augmentation were obtained based on manual experience, and such inclusion of human intervention might easily introduce cognitive biases and manual errors. For this reason, we combined the previously described MTPI and reinforcement learning (Watkins and Dayan, 1992; Russo et al., 2018) [e.g., TSA (Russo et al., 2018; Dai et al., 2022)] to automatically determine the optimal processing methods and parameter ranges in the data augmentation process. The TSA is a common reinforcement learning algorithm for solving the optimal strategy problem for intelligence in DTV.

Since the data auto-augmentation process involves two-dimensional strategy problems with different processing method processes (M) and different value ranges (N), the traditional TSA was improved to MTSA (Multi-Thompson Sampling algorithm) with multiple levels of inputs for implementation ($M \times N$). Where the inputs can be represented as an input matrix Q , as represented in (Equation 9).

$$Q = [Q_1, \dots, Q_M] = \begin{bmatrix} q_{1,1} & \dots & q_{M,1} \\ \vdots & q_{n,m} & \vdots \\ q_{1,N} & \dots & q_{M,N} \end{bmatrix} \quad (9)$$

where, m is the number of augmentations method, with $m = 1, \dots, M$, e.g., $M = 10$ for this study. Where $m = 1$ denotes a random reduction with a magnification of 0 to 1, $m = 2$ denotes a random enlargement with a magnification of larger than 1, $m = 3$ denotes an inverse rotation with a rotation angle of less than 0, $m = 4$ denotes a forward rotation with a rotation angle of larger than 0, $m = 5$ denotes left-biased horizontal cropping, $m = 6$ denotes right-biased horizontal cropping, $m = 7$ denotes vertical cropping in the downward direction, $m = 8$ denotes vertical cropping in the upward direction, $m = 9$ denotes horizontal translation in the left direction, and $m = 10$ denotes horizontal translation in the right direction. N is the value range of each column matrix $Q_m = [q_1, \dots, q_N]^T$ which is evenly partitioned into N value accuracies, in this study, it is assumed to use a variable with 8-bit storage space to represent the value possibilities, thus $N = 256$ (denoted as $n = 1, 2, \dots, 256$ or $n = 0, 1, 2, \dots, 255$). In

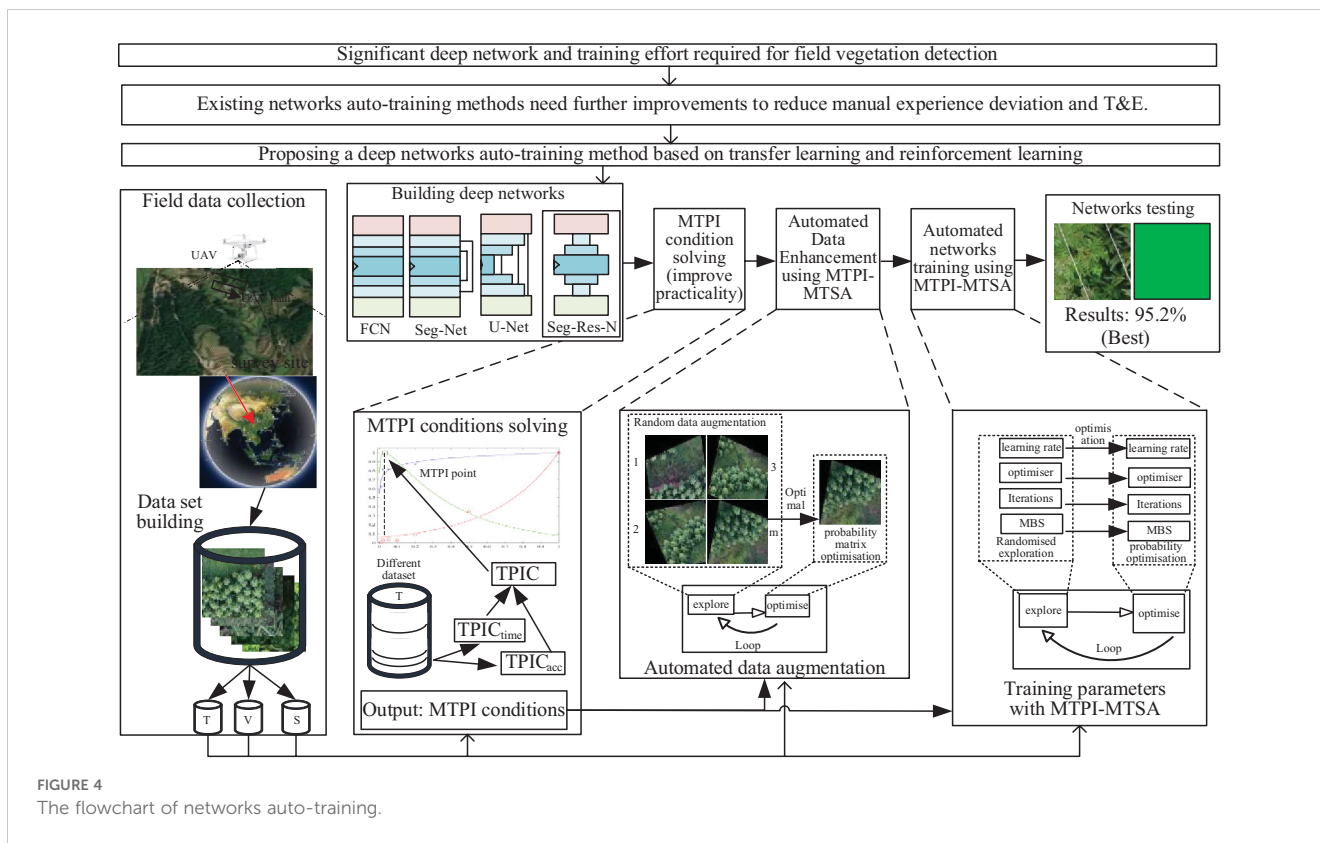


FIGURE 4 The flowchart of networks auto-training.

MTSA, the probability of each column matrix Q_m is calculated separately. It is denoted as (Equation 10).

$$P(q_{n,m}) = q_m / \sum Q_m \tag{10}$$

Before iteration, all elements of Q_m were set to 1 to avoid a zero denominator. During each iteration, the corresponding element of Q_m that achieves the maximum result will be updated to bias the MTSA results towards achieving the best value, as (Equation 11).

$$q_{n,m}^{t+1} = q_{n,m}^t + r \times \log(t) \tag{11}$$

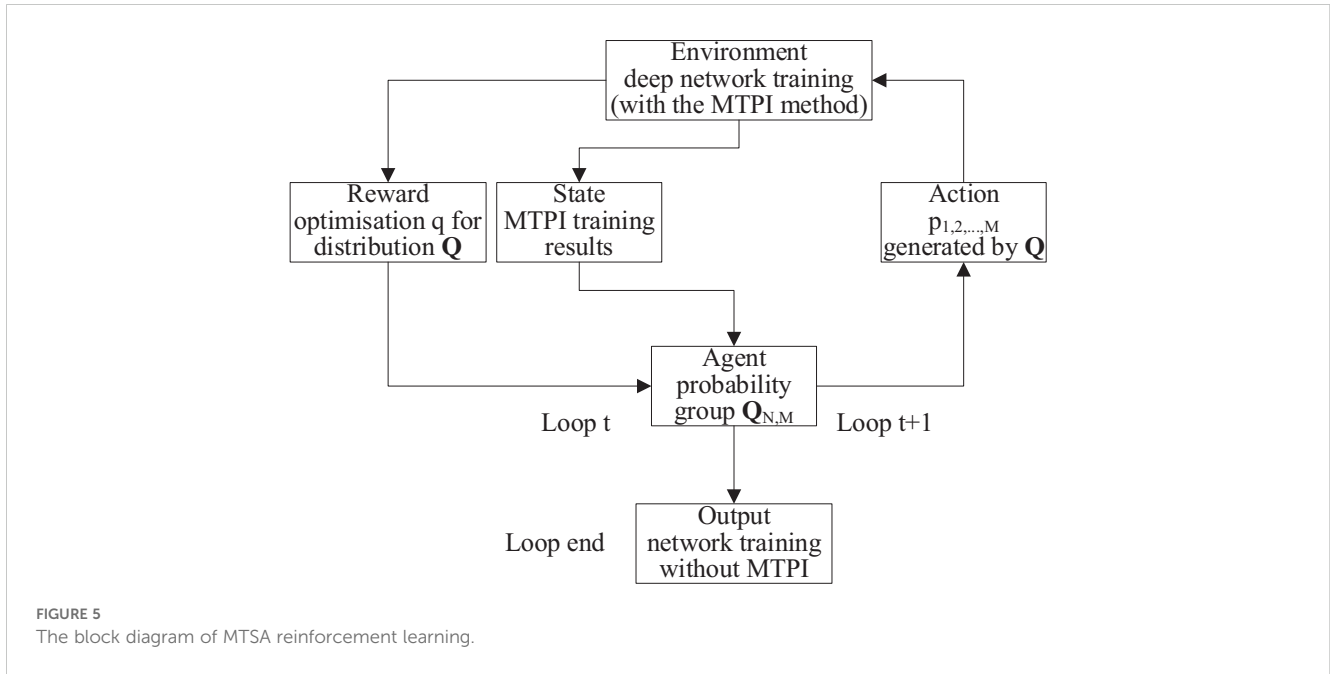
Where r is the learning rate, e.g., $r = 4$. The t is the number of MTSA iterations. In addition, $q_{n,m}$ also sets upper and lower limits to prevent errors due to overflow of computer digits, e.g., 10^{-10} and 10^{10} . With Q renewing, the data augmentation process and the values ranges will move towards biasing the optimal feedback (the network accuracy) to obtain the optimal solution (or near-optimal solution).

Because of the manual data augmentation methods usually use the maximum processing and commonly available parameters (e.g., multiples of 10). Therefore, to gain simulated manual method results for comparison, all the involved processing and five common parameters were taken for data augmentation, including $mt = -5, -4, \dots, 5$. During the test of data augmentation, mt would be mapped to actual values. Where, the reduction is $(1 - 10^{mt})$, the enlargement is $1/(1 - 10^{mt})$, the rotation angle is $\pm [45 \times 10^{mt}]$, and the offset or crop value is $\pm [50 \times 10^{mt}]$, the $[\]$ means rounding up to the nearest integer.

Figure 5 shows the schematic of MTSA reinforcement learning in this paper, including the interactions of the agent, environment, state, action, reward, and output. The agent performs a $t+1$ action according to the policy (Equation 11) based on the t state and the t reward feedback from the environment. The Agent is a hypothetical individual acting in the environment, and the actions it produces in the environment are controlled by a probability distribution Q that can be iteratively optimized. In the $t+1$ loop, the $t+1$ action of the agent consists of M sub-actions of $p_{1,2,\dots,M}$. The action $p_{1,2,\dots,M}$ is transitioned to a $t+1$ state in the environment, and a $t+1$ reward positive or negative reward) is generated. The output of the environment (means the agent in environment) is the MTPI training result corresponding to the action, and although each action is retrained in multiple cycles, the time consumption of the MTPI has been significantly reduced, and the whole reinforcement learning process is still acceptable. After the maximum number of loops is reached, the result output by the agent is used as the auto-training parameter of the network (the output result is without MTPI).

3.3 The MTPI based reinforcement learning for deep network auto-training

The MTPI-MTSA introduced in section 3.2 is an effective automatic method to replace manual experience, so it is similarly used in the DL auto-training. In the implementation of MTPI-MTSA for network auto-training, the mapping form of the input



matrix was optimized to make the inputs match the auto-training. The cost function of MTPI-MTSA was also optimized so that the output considers the result accuracy and the time consumption factor would not be ignored at the same time.

As reported in scholarly studies, the training parameters affecting the DL accuracy and time consumption include, OT (Optimizer Types) (Elshamy et al., 2023), LR (Learning Rate) (Lewkowycz et al., 2020), IN (Iterations Number) (Xiang et al., 2021), and MBS (Mini Batch Size) (Kandel and Castelli, 2020). Therefore, the input matrix Q of the MTPI-MTSA contains the four-parameter dimensions mentioned, denoted as $M = 4$ and $N = 256$ to represent them during the auto-training. Among them, $m = 1$ represents the OT, and the optimizer introduced in this paper included “sgdm” (Stochastic Gradient Descent with Momentum), “Adam” (Adaptive Moment Estimation), or “RMSprop” (Root Mean Square Propagation) which was shown as (Equation 12).

$$OP = \begin{cases} 1, & q_1 < N/3 \\ 2, & N/3 \leq q_1 < 2N/3 \\ 3, & 2N/3 \leq q_1 \end{cases} \quad (12)$$

Where m is the ordinal number of q_m , $OP = 1$ means the OT is “sgdm” optimizer, $OP = 2$ means the “Adam” optimizer, and $OP = 3$ means the “RMSprop” optimizer. In addition, $m = 2$ denotes the LR and the mapping relationship was given in (Equation 13).

$$LR = 10^{-1 \times (l+r \times q_2/N)} \quad (13)$$

Where l and r are the two limiting factors of the LR value range, such as $l = 1$ and $r = 6$. Thus, the value range of LR is $[10^{-7}, 10^{-1}]$. And then $m = 3$ denoted the max epochs of IN, and it will be rounded to the nearest integer. When $m = 4$, the MBS with the value range (1,256), its upper bound can be changed according to the GPU RAM, but must be an integer.

To improve the MTPI-MTSA on network accuracy and time consumption to make deep training automatic, a time consumption component is added to the cost function, as shown in (Equation 14).

$$CF = A \times t/T \quad (14)$$

Where A is the accuracy, T is the relative iteration number (relative time consumption), and t is the iteration number where the minimum loss was obtained in the training process. Since A has a higher priority relative to T , the T will be considered only when $A \geq A_0$, while the A_0 can be set as the MTPI result in section 3.1 and section 3.2. Where T can be obtained as (Equation 15).

$$T = t/t_0 \times 100\% \quad (15)$$

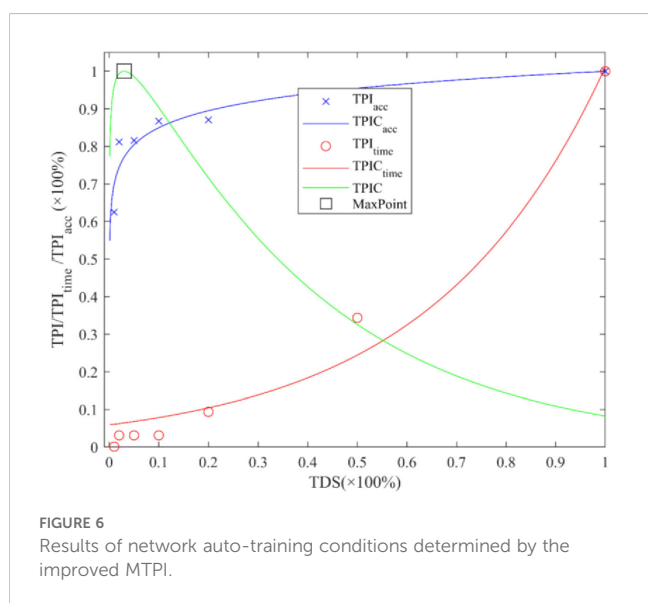
To further validate and test the effectiveness of the complete network auto-training, the data auto-augmentation and network auto-training were performed on the structures of FCN, Seg-Net, U-Net, and Seg-Res-Net 50 built-in section 2.2. Where the data auto-augmentation was performed in the MTPI-MTSA method introduced in section 3.2 and section 4.2 as AD . The four networks were auto-trained with the methodology presented above and were tested with the SD introduced in section 2.1.

4 Results and analysis

4.1 Results of the improvement MTPI for deep learning auto-training

The Figure 6 shows the results of network auto-training conditions (the MTPI conditions) determined by the improved MTPI in DTV environment. The graph, including the TPI_{acc} index (when $TDS = 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1$), the TPI_{time} , the fitting curve of $TPIC_{acc}$ [when $TDS \in (0, 1)$], the fitting

curve of $TPIC_{time}$, the TPIC index curve, and the value of MTPI (the maximum point), shows the impact of the TDS value on the network accuracy and time cost. The trends of $TPIC_{acc}$ and $TPIC_{time}$ showed that the potential indices of accuracy and time spent would both increase as the TDS increases; but the rising trend of $TPIC_{acc}$ keeps decreasing, while the rising trend of $TPIC_{time}$ keeps increasing. This might be due to the TDS influence decreasing as the network gets closer to the optimal when its training process gets more difficult; but the effect of TDS on accuracy is still controversial, (conjecture that under field vegetation data conditions TDS can be close to a logarithmic relationship with the network performance). In addition, the relationship of TDS on $TPIC_{time}$ (training time consumption) is approximately exponential, which may be due to TDS increase affecting not only the batch number but also the final network performance when training in MBS (mini-batches size). The TPIC trend is increasing and then decreasing which is consistent with the trends of $TPIC_{acc}$ and $TPIC_{time}$ analyzed earlier, which might be because $TPIC_{acc}$ (the numerator of the TPI obtaining formulae) increases faster when TDS is small whereas $TPIC_{time}$ (the denominator of the TPI) increases slowly; thereafter, the $TPIC_{acc}$ increases at a reduced rate and $TPIC_{time}$ increases at an elevated rate. It also shows that, when the TDS is small, an increase in TDS can quickly improve the network performance without much change in time consumption. However, when the TDS is already large, a further increase in TDS has less impact on the network performance but consumes a huge amount of time, which is consistent with the phenomenon that the performance enhancement of DL slows down gradually in network training. The value corresponding to the TPIC maximum (maximal point) is the result of the MTPI method (MTPI condition, including TDS and LT). Obviously, the MTPI condition is not the performance-optimal condition for networks, but this condition is the best-combined combination of performance and time cost, which is beneficial to discuss the improvement of DL and optimization of parameters.



More specifically, the fitted curve $TPIC_{acc}$ has the eq. $TPIC_{acc} = a \times \ln(TDS) + b$, where $a = 0.0650$ and $b = 0.995$, the $RMSE = 0.0485$ and the $R^2 = 0.858$. The fitted equation for the fitted curve $TPIC_{time}$ is $TPIC_{time} = a \times e^{(b \times TDS)}$, where $a = 0.0591$ and $b = 2.84$; the $RMSE = 0.0604$ and $R^2 = 0.977$. Thus, the MTPI conditions obtained are $TDS = 2.30\%$, $LT = 6.31\%$, and the reference accuracy $WPA_{ref} = 76.3\%$.

It can be seen that with the introduction of the MTPI method, the optimization process requires only 2.30% dataset size and 6.31% time-consuming. The MTPI has largely improved the utility and adaptability for later network auto-training. In this study, we investigate the transfer training results of the same network [trained in a detection task (He et al., 2016)] transferred to the same dataset with different parameter conditions, while the previous paper investigated the transfer training of a network on a different dataset, but we considered such results still reasonable. The reason might be that the starting point of TL (He et al., 2016) has obtained some basic primitive features (e.g., edges or corner points), and all these features have a strong role in the detection of the image. However, as can be seen from the starting point of the curve in the figure (e.g., $TDS < 0.05$), the network without TL has a relatively large number of non-universal features (lower accuracy, < 0.6). This result also reflects, to a certain extent, the superiority of the MTPI algorithm, which aims to achieve speedups under the premise of obtaining the most efficient results with the smallest TDS.

4.2 The results of auto dataset augmentation with MTPI-MTSA

Figure 7 gives the output curve as iterations of the MTPI-MTSA during the dataset auto-augmentation of the optimal solution. This contains the red result curve (accuracy, WPP), the blue reference curve (obtained from the MTPI method above), the green 10 t (times of iteration) average, and the 10-t mean-error curve (10 t standard deviation of the green curve). The red curve shows that the algorithm gets poor ($< 55\%$) results in the early stage (≤ 50 t), changes significantly in the middle stage (> 50 and ≤ 100 t), and shows a high result and small changes in the later stage (> 100 t). This resultant performance corresponds to the mean-error curve, with low mean scores and low errors in the early part, then the mean scores increase in mean but with higher errors in the middle part, and the mean scores stabilize at a larger value (compared to the reference line) with low errors in the late part. These results are also consistent with the TSA which has phases of self-exploration and convergence to the best value. The MTPI-MTSA falls back near the later stages (e.g., 110–130 t periods), which may be due to possible unfavorable exploration in the MTSA iterations, but stabilizes at a higher level in the later stages as the algorithm converges towards better results, which may be due to the late algorithmic effect of the algorithm approaching the optimal solution. This also suggested that the relationship between data enhancement variables and network accuracy for MTPI-MTSA in field vegetation data is

likely not a probability distribution in the traditional sense (e.g., normal or beta).

Table 2 shows the data augmentation methods and parameters obtained by the proposed MTPI-MTSA, including Reduce Scale, Enlarge Scale, Rotate, Offset Left/Right, Crop Left/Right, Crop Up/Down, and Offset Up/Down. The table shows that the scale range between 0.81 and 1.64 is an optimal value (or near-optimal value), which may be related to the UAVS and DTV terrain factors. Although the UAVS is operating at 200 m altitudes, the DTV has a large difference in horizontal altitude and terrain, so it is necessary to have some image scale optimization to achieve the differentiation. In addition, the offset and cropping of the images are concentrated around 23 (between 18 and 26, which is about less than 10%, with an image scale of 250), probably because the offset and cropping have similar linear geometric transformations (concentrated in constant terms in the image transformation matrix). However, they still have some data errors, which is possible because the optimization process of MTPI-MTSA is stochastic and the error in the results is the stochastic optimization process. Such results with some errors are acceptable, firstly because the subsequent auto-training process can accept a certain range of data augmentation results, and secondly because such data augmentation with variation can lead to more diverse stochastic data for training.

The results of the manual method show that when $mt = -5, -4, \dots, 5$ $WPA = 73.05, 72.54, 73.12, 69.61, 14.56, 73.70, 73.11, 73.42, 68.54, 26.13(\%)$, with a mean value of 60.30% and a std. var. of 20.94 ($ref. = 76.3(\%)$). Comparing the manual and automated method results shows that the optimal results of the manual method can be close to the reference value, but the manual method has a T&E cost (std. > 0), which explains that the manual method needs multiple tests and parameter debugging to obtain the best. The introduced MTPI-MTSA automatically obtained the optimal results (or near-optimal solutions) with no human involvement or T&E process, which also proves that the MTPI-MTSA is necessary for fully automated data augmentation of network training. The results showed that the manual method was worse than the automated and it might be because the

TABLE 2 Results of the dataset augmentation method obtained with the MTPI-MTSA.

Methods	Lower limit	Upper limit
Reduce/Enlarge Scale	0.81	1.64
Rotate	-25.88	36.08
Offset Left/Right	-	-
Crop Left/Right	-21.96	19.80
Crop Up/Down	-25.88	18.82
Offset Up/Down	-	-

manual method used all processing methods and uniform parameters in this study, but not all of them were suitable for vegetation segmentation in DTV environments.

4.3 The auto-training results of deep networks with MTPI-MTSA

Figure 8 shows the networks auto-training results with the MTPI-MTSA, including the iterative results curve (WPP) in red, the reference curve in blue (derived from the improved MTPI), the 10-iteration average curve and the 10-iteration mean-error curve in green (standard deviation for error). Similar to section 4.2 the iterative optimization process can be divided into three stages, a pre-stage (≤ 50 t), a middle stage (> 50 and ≤ 150 t), and a late stage (> 150 t). It can be seen that the resulting curve and the average curve gradually increase from the lowest point but contain a large error in the early stage, which is consistent with the TSA has a certain self-exploration and tends to the optimal law, and the results are also consistent with the results in section 4.2. In addition, the output and average curves showed a high output with a gradually decreasing error in the middle stage, which indicates the algorithm score can reach a larger score position and gradually smooth out. Although the process can still be unfavorable exploration and lead to a certain degree of fluctuation in the output curve, as the iterations increase, the probability of unfavorable exploration is gradually reduced. Finally, both the result curve and the mean curve are at higher curves and the error is small in the late stage. The mean value of the score increases but also has a larger error in the middle of the algorithm, and stabilizes at a larger value (above the reference line, blue line) and with a smaller error in the later stages. Such curve performance indicates auto-training, it may be due to the structure of the MTPI-MTSA used and the fact that the optimization training in the DTV environment has a more concentrated high probability of taking values (Compared with trends in data auto-augment above).

To further demonstrate the network auto-training method proposed in this paper, the established Seg-Net, FCN, U-Net, and Seg-Res-Net 50 network structures were similarly processed and compared. Three sample images were randomly selected for

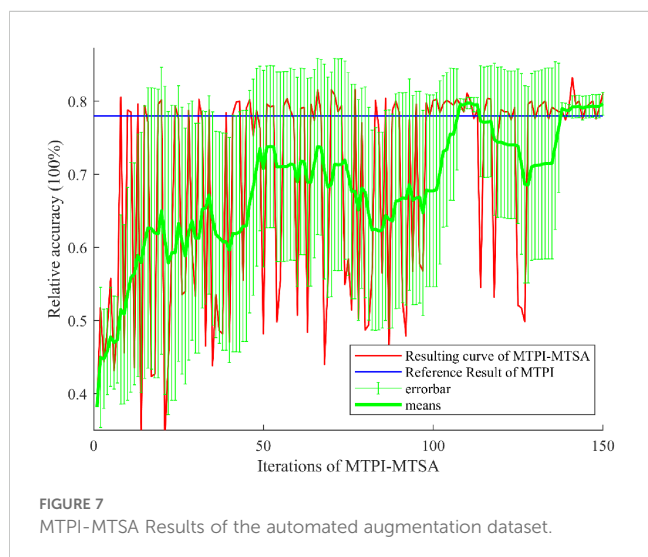
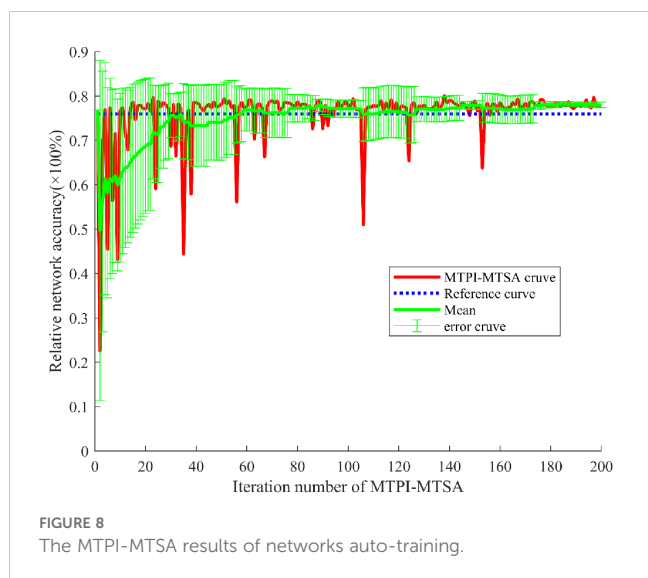


FIGURE 7 MTPI-MTSA Results of the automated augmentation dataset.

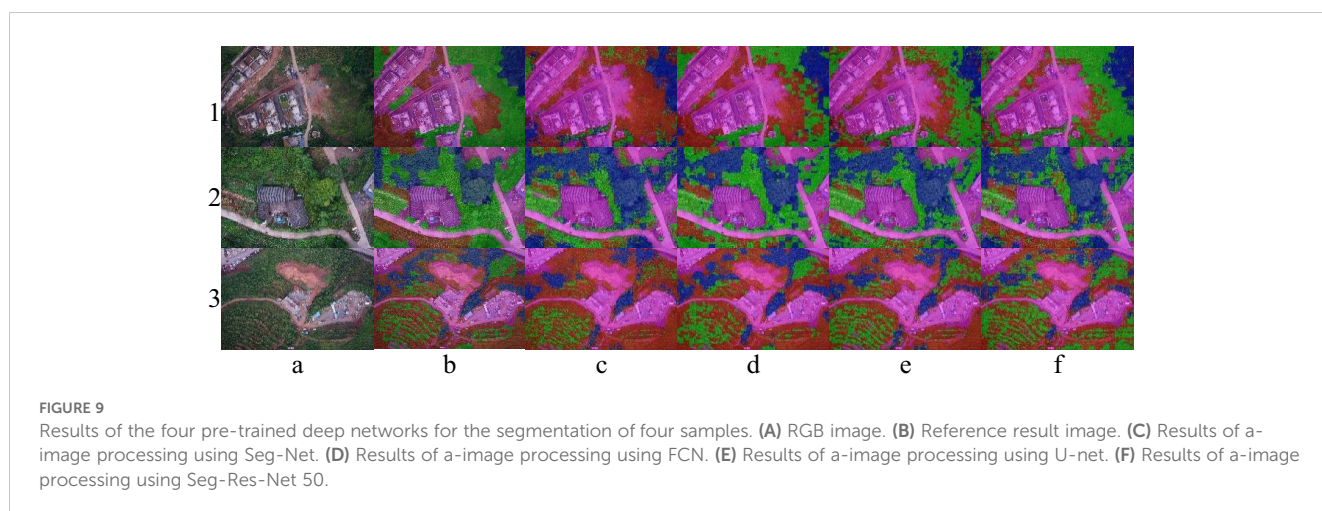


processing and testing respectively, which have been given in Figure 9. In the labeled RGB image results, blue, red, green, and magenta markers with 20% transparency were used to indicate the segmented regions of TR, SR, GR, and NVR, respectively. Column a in each row represents the original RGB image, column b represents the manually labeled results (ref.), and columns c, d, e, and f represent the network output. Overall, the four networks achieve vegetation segmentation to some extent, e.g., all four networks achieve good results in the NVR compared to column b. It might show that the segmentation task in the NVR is the easier part, when the NVR features are visible and obvious (e.g., color and texture feature), even easy for the human eye. In addition, the U-Net seems to treat the SR part more as TR, e.g., Figure 9 1. e (1. e denotes the resultant image in row 1, column e), while the FCN misclassifies it more as GR, e.g., Figure 9 2. e, which may be related to the convolution depth of the network, with deeper networks having better resultant performance. However, Seg-Res-Net 50 still

significantly outperforms U-Net in the SR region, as shown by the difference between 3. e and 3. f in Figure 9, which may be related to the performance structure. In these results, the segmentation of Seg-Res-Net 50 is the best among all vegetation species, which may be related to the network residual structure introduced in Seg-Res-Net 50, which is consistent with the results of Alfanindya, A. (Alfanindya et al., 2013). and Yu, H. (Yu et al., 2021).

Figure 10 shows the confusion matrix results (an overall statistical result of a dataset) for the four networks on the SD (testing set, introduced in section 2.1). The horizontal axis was the pixel point mapping output results for the categories, while the vertical axis represents the actual category, i.e., the first column on the horizontal axis corresponds to the TR, SR, GR, and NVR. The diagonal elements of the confusion matrix from the top left to the bottom right reflect the network accuracies for the four categories, with higher values showing better network performance. The results showed that all four networks achieved some performance on the dataset, with Seg-Res-Net 50 achieving the highest overall accuracy (97.60% in Figure 10D), while Seg-Net showed the lowest overall accuracy (64.36% in Figure 10A). The results in Figure 10 showed that Seg-Net had a weak segmentation performed for SR and GR (Figure 10A, only about 64.36 and 83.57%), FCN is slightly better than that of Seg-Net for GR and SR (about 10% improvement on SR, Figure 10B), on the other hand, Seg-Res-Net 50 was better than the three previous network structures in all categories (Figure 10D). This result may be related to the residual connection structure in Seg-Res-Net 50 since the residual network was allowed to learn deeper structures efficiently without being prone to overfitting, which is consistent with the earlier findings.

Table 3 shows the results of the four auto-trained networks on the SD from five perspectives (WPA, WPP, WRE, WF1, and WIoU). Among them, WPA is the overall prediction accuracy (intuitive performance); WPA is the accuracy of the positive category prediction (positive category accuracy); WRE is the coverage for the actual positive category (positive category



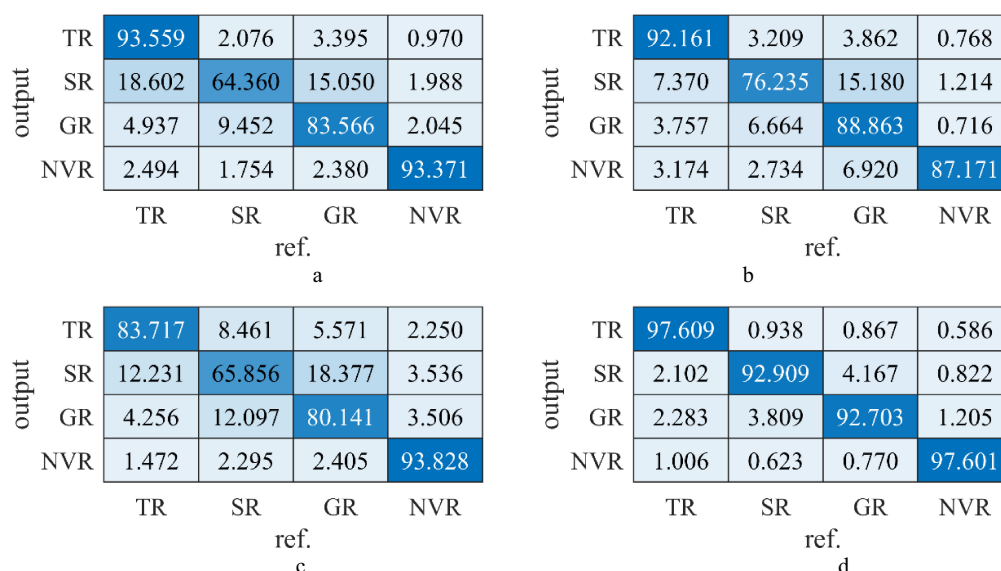


FIGURE 10 Confusion matrix results of the four auto-trained networks on the SD. (A) Confusion matrix of FCN. (B) Confusion matrix of Seg-Net. (C) Confusion matrix of U-Net. (D) Confusion matrix of Seg-Res-Net 50.

detection); WF1 is the evaluation of the network performance (a combination of WPA and WRE); and WIoU is the overlap between the actual and predicted regions. Based on the above five evaluation perspectives, Seg-Res-Net 50 was found the best overall performance than Seg-Net, U-Net, or FCN, (e.g., the WIoU in Table 3). These results may have been attributed to the fused residual structure of the Seg-Res-Net 50 with deeper network layers and better vegetation segmentation capabilities. Combining the above example analysis (Figure 10) and the statistical results analysis (Table 1), the auto-trained Seg-Res-net 50 outperformed the other three networks in terms of overall segmentation effect and per-class segmentation effect. This also indicated that Seg-Res-net 50 performed well on simple tasks (NVA) and complex tasks (GR, SR, and TR), both in detail and overall. It also demonstrates that the proposed auto-training method, based on MTPI and reinforcement learning, can be used

in complex network structures (e.g., Seg-Res-Net 50) and complex dataset environments (e.g., DTV in the field).

To verify the advantages of the proposed network auto-training method in this manuscript, four similar methods are given in Table 4 for comparative discussion. The evaluation perspectives include dataset size, data sample size, network structure, acceleration methods, and acceleration effectiveness. From the two perspectives of dataset size and sample size (the two columns of the table), the proposed method has a larger dataset, and the object of the dataset is the vegetation image data in an unstructured DTV environment, which can be seen that the proposed method has a higher degree of compatibility. In addition, the network structure in Table 4 shows the proposed method is more adaptable to complex networks (comparing the network depth of the proposed method and the previous three methods). Although Wang, S.'s method has a more complex structure, his paper does not include the process of automatically determining the optimal training parameters, which might require the algorithm to repeatedly confirm the optimal conditions. This result also implies more manual setup experience for implementation, and it further surfaces the greater practicality of the proposed method. Finally, the acceleration methods and acceleration effects are also compared, which shows the potential future of the overall method. Among them, the acceleration means of the proposed study, MTPI, improves the overall iteration speed to a great extent, making the proposed auto-training method very promising for future applications.

TABLE 3 Results of five evaluation factors for the four network structures (%).

Networks	WPA	WPP	WPR	WPF1	WIoU
FCN	83.714	84.023	83.714	83.395	72.283
Seg-Net	86.108	86.691	86.108	86.142	75.940
U-Net	80.886	80.704	80.886	80.706	68.463
Seg-Res-Net 50	95.205	95.201	95.205	95.195	90.881

TABLE 4 Performance differences compared to three existing methods.

Methods	Dataset size	Sample size	Networks	Acceleration method	Time cost
Proposed method	30,000	3×250 ²	Res-Net 50 (58) ^a	MTPI	8.7%
Sun, Y. method (Sun et al., 2020)	6,000	3×32 ²	Auto-CNN (10)	- ^b	100%
Xiao, X. method (Xiao et al., 2020)	6,000	3×32 ²	CNN (14)	early stop	75%
Montes, C. Method (Montes et al., 2023)	220,000	1×128	CNN (10)	early stop	50%
Wang, S. method (Wang S. et al., 2020)	1266	1×512 ²	DenseNet121(>60)	-	-

^aThe number in “()” means layers number, such as the “Seg-Res-Net 50 (58)” means the Seg-Res-Net 50 has 58 layers. In this paper, the network layers number is denoted by the number of active layers.

^bThe “-” means used no time-saving method.

5 Conclusion and discussion

To reduce the human intervention and T&E cost for deep network training as well as to enhance the existing methods' practicality and adaptability, an auto-training method was proposed based on the MTPI and MTSA in the DTV environment. The main conclusions are as follows.

1. The MTPI was improved to automatically calculate the most suitable transfer conditions and found that the subsequent automation only requires 2.30% data and 6.31% time, which is adaptable to the complex networks in DTV.
2. The MTPI conditions were incorporated into the automatic dataset augmentation prior to network training and found a 20.94% T&E reduction and a 16.00% accuracy improvement.
3. The auto-training method was used for four common networks (e.g. FCN, Seg-Net, U-Net, Seg-Res-Net 50) in a DTV environment, and the best Seg-Res-Net 50 achieved 93.6% WPA and 87.6% WIoU. The proposed auto-training is more practical than several existing methods, and avoids human intervention, providing a reference for automated deep learning applications in unstructured environments.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

YC: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. BZ: Methodology,

Formal analysis, Writing – review & editing. XC: Writing – review & editing. CM: Writing – review & editing. LCu: Writing – review & editing. FL: Writing – review & editing. XH: Writing – review & editing. SW: Writing – review & editing. LCh: Writing – review & editing. DY: Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. Supporting projects of this paper: the National Natural Science Foundation of China (Grant Nos. 52222903; 41977007; 52409067), Scientific research project supported by China Three Gorges Construction Man-agreement Co., LTD (BHT/0869), and Modern Ecological Irrigation District Construction Model based on Rigid Constraints on Water Resources, Xinjiang Autonomous Region, Sub-title of Major Science and Technology Special Projects (2023A02002-2).

Conflict of interest

FL was employed by Central South Survey and Design Institute Group Co., Ltd. XH was employed by China Electric Construction Group Beijing Survey and Design Institute Co.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Abayomi-Alli, O. O., Damaševičius, R., Misra, S., and Maskeliūnas, R. (2021). *Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning* (USA: Wiley - Blackwell Publishing, Inc) 38. doi: 10.1111/exsy.12746
- Alfanindya, A., Hashim, N., and Eswaran, C. (2013). "Content Based Image Retrieval and Classification using speeded-up robust features (SURF) and grouped bag-of-visual-words (GBoVW)," in *2013 International Conference on Technology, Informatics, Management, Engineering and Environment*, Bandung, Indonesia. (IEEE) 2013, pp. 77–82. doi: 10.1109/TIME-E.2013.6611968
- Al-Hyari, A., and Abu-Faraj, M. (2022). "Hyperparameters optimization of convolutional neural networks using evolutionary algorithms," in *2022 International Conference on Emerging Trends in Computing and Engineering Applications (ETCEA)*, Karak, Jordan (ETCEA) 2022, 1–6. doi: 10.1109/ETCEA57049.2022.10009778
- Ali, M., Borgo, R., and Jones, M. W. (2021). Concurrent time-series selections using deep learning and dimension reduction. *Knowledge-Based Syst.* 233, 107507. doi: 10.1016/j.knsys.2021.107507
- Ang, K. L. M., and Seng, J. K. P. (2021). Big data and machine learning with hyperspectral information in agriculture. *IEEE Access* 9, 36699–36718. doi: 10.1109/Access.6287639
- Ayhan, B., Kwan, C., Budavari, B., Kwan, L., Kwan, L., Lu, Y., et al. (2020). Vegetation detection using deep learning and conventional methods. *Remote Sens.* 12, 2502. doi: 10.3390/rs12152502
- Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 2481–2495. doi: 10.1109/TPAMI.34
- Chen, L., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). *Encoder-decoder with atrous separable convolution for semantic image segmentation* (Cham: Springer International Publishing).
- Chen, X., Chen, W., Dinavahi, V., Liu, Y., and Feng, J. (2023). Short-term load forecasting and associated weather variables prediction using resNet-LSTM based deep learning. *IEEE Access* 11, 5393–5405. doi: 10.1109/ACCESS.2023.3236663
- Chen, Y., Zhou, B., Ye, D., Cui, L., Feng, L., and Han, X. (2023). An optimization method of deep transfer learning for vegetation segmentation under rainy and dry season differences in a dry thermal valley. *Plants (Basel)* 12, 3383. doi: 10.3390/plants12193383
- Dai, Z., Shu, Y., Low, B. K. H., and Jallet, P. (2022). Sample-then-optimize batch neural thompson sampling. *ArXiv.* 35, 23331–23344. doi: 10.48550/arXiv.2210.06850
- Ding, F., Li, C., Zhai, W., Fei, S., Cheng, Q., and Chen, Z. (2022). Estimation of nitrogen content in winter wheat based on multi-source data fusion and machine learning. *Agriculture* 12, 1752. doi: 10.3390/agriculture12111752
- Elshamy, R., Abu-Elnasr, O., Elhoseny, M., and Elmougy, S. (2023). Improving the efficiency of RMSProp optimizer by utilizing Nesterov in deep learning. *Sci. Rep.* 13, 8814. doi: 10.1038/s41598-023-35663-x
- Fritsch, F. B. (2020). Crop monitoring using satellite/UAV data fusion and machine learning. *Remote Sens.* 12 (9), 1357. doi: 10.3390/rs12091357
- Gao, J., Li, P., Chen, Z., and Zhang, J. (2020). A survey on deep learning for multimodal data fusion. *Neural Comput.* 32, 829–864. doi: 10.1162/neco_a_01273
- García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., and Herrera, F. (2016). Big data preprocessing: methods and prospects. *Big Data Analytics* 1, 9. doi: 10.1186/s41044-016-0014-0
- Gentilhomme, T., Villamizar, M., Corre, J., and Odobez, J. (2023). Towards smart pruning: ViNet, a deep-learning approach for grapevine structure estimation. *Comput. Electron. Agric.* 207, 107736. doi: 10.1016/j.compag.2023.107736
- Hassabis, D., Kumaran, D., Summerfield, C., and Botvinick, M. (2017). Neuroscience-inspired artificial intelligence. *Neuron* 95, 245–258. doi: 10.1016/j.neuron.2017.06.011
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). *Delving deep into rectifiers: surpassing human-level performance on imageNet classification* (Santiago, Chile: IEEE Computer Society).
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, 770–778. doi: 10.1109/CVPR.2016.90
- Hu, X., Xie, C., Fan, Z., Duan, Q., Zhang, D., Jiang, L., et al. (2022). Hyperspectral anomaly detection using deep learning: A review. *Remote Sens.* 14 (9), 1973. doi: 10.3390/rs14091973
- Kamarudin, M. H., Ismail, Z. H., and Saidi, N. B. (2021). Deep learning sensor fusion in plant water stress assessment: A comprehensive review. *Appl. Sci.* 11, 1403. doi: 10.3390/app11041403
- Kandel, I., and Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express, published by the Korean Institute of Communications and Information Sciences (KICS)*.
- Kang, H., and Wang, X. (2023). Semantic segmentation of fruits on multi-sensor fused data in natural orchards. *Comput. Electron. Agric.* 204, 107569. doi: 10.1016/j.compag.2022.107569
- Lewkowycz, A., Bahri, Y., Dyer, E., Sohl-Dickstein, J., and Gur-Ari, G. (2020). The large learning rate phase of deep learning: the catapult mechanism. *ArXiv.* 2003.02218. doi: 10.48550/arXiv.2003.02218
- Li, W., Buitenwerf, R., and Munk, P. (2020). Deep-learning based high-resolution mapping shows woody vegetation densification in greater Maasai Mara ecosystem. *Remote Sens. Environment: Interdiscip. J.* 247 (15), 111953. doi: 10.1016/j.rse.2020.111953
- Li, D., Song, Z., Quan, C., Xu, X., and Liu, C. (2021). Recent advances in image fusion technology in agriculture. *Comput. Electron. Agric.* 191, 106491. doi: 10.1016/j.compag.2021.106491
- Lin, S., Xiu, Y., Kong, J., Yang, C., and Zhao, C. (2023). An effective pyramid neural network based on graph-related attentions structure for fine-grained disease and pest identification in intelligent agriculture. *Agriculture.* 13, 567. doi: 10.3390/agriculture13030567
- Liu, Y., Jiang, C., Lu, C., Wang, Z., and Che, W. (2023). Increasing the accuracy of soil nutrient prediction by improving genetic algorithm backpropagation neural networks. *Symmetry* 15, 151. doi: 10.3390/sym15010151
- Long, J., Shelhamer, E., and Darrell, T. (2015). *Fully convolutional networks for semantic segmentation* (Ithaca: Cornell University Library).
- Lu, S., Song, Z., Chen, W., Qian, T., Zhang, Y., Chen, M., et al. (2021). Counting dense leaves under natural environments via an Improved deep-learning-based object detection algorithm. *Agriculture.* 11, 1003. doi: 10.3390/agriculture11101003
- Mandal, B., Okeukwu, A., and Theis, Y. (2021). *Masked face recognition using resNet-50* (Ithaca, New York: ArXiv).
- Marzougui, A., Mcgee, R., Vleet, S. V., and Sankaran, S. (2023). Remote sensing for field pea yield estimation: A study of multi-scale data fusion approaches in phenomics. *Front. Plant Sci.* 14. doi: 10.3389/fpls.2023.1111575
- Montes, C., Morehouse, T., Kasilingam, D., and Zhou, R. (2023). "Optimized CNN auto-generator using GA with stopping criterion: design and a use case," in *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, 2023, 1009–1014. doi: 10.1109/CCNC51644.2023.10060860
- Mu, C., Guo, Z., and Liu, Y. (2020). A multi-scale and multi-level spectral-spatial feature fusion network for hyperspectral image classification. *Remote Sens.* 12, 125. doi: 10.3390/rs12010125
- Nasir, I. M., Bibi, A., Shah, J. H., Khan, M. A., Sharif, M., Iqbal, K., et al. (2021). Deep learning-based classification of fruit diseases: an application for precision agriculture. *Computers Materials Continua.* 66, 1949–1962. doi: 10.32604/cmc.2020.012945
- Ni, H., Shi, Z., Karungaru, S., Lv, S., Li, X., Wang, X., et al. (2023). Classification of typical pests and diseases of rice based on the ECA attention mechanism. *Agriculture.* 13, 1066. doi: 10.3390/agriculture13051066
- Nikolados, E., Wongprommoon, A., Aodha, O. M., Cambray, G., and Oyarzún, D. A. (2022). Accuracy and data efficiency in deep learning models of protein expression. *Nat. Commun.* 13, 7755. doi: 10.1038/s41467-022-34902-5
- Niu, Z., Zhong, G., and Yu, H. (2021). A review on the attention mechanism of deep learning. *Neurocomputing* 452, 48–62. doi: 10.1016/j.neucom.2021.03.091
- Otoni, A. L. C., Amorim, R. M., Novo, M. S., and Costa, D. B. (2023). Tuning of data augmentation hyperparameters in deep learning to building construction image classification with small datasets. *Int. J. Mach. Learn. Cybernetics* 14, 171–186. doi: 10.1007/s13042-022-01555-1
- Patil, R. R., and Kumar, S. (2022). Rice-fusion: A multimodality data fusion framework for rice disease diagnosis. *IEEE Access* 10, 5207–5222. doi: 10.1109/ACCESS.2022.3140815
- Pu, W., Wang, Z., Liu, D., and Zhang, Q. (2022). Optical remote sensing image cloud detection with self-attention and spatial pyramid pooling fusion. *Remote Sens.* 14, 4312. doi: 10.3390/rs14174312
- Reyer, D. (2022). Comparison of autoML solutions auto-pyTorch and autoCVE. *Ontology based Meta AutoML.* doi: 10.48444/h_docs-pub-270
- Ronneberger, O., Fischer, P., and Brox, T. (2015). *U-net: convolutional networks for biomedical image segmentation* (Cham: Springer International Publishing).
- Rosende, S. B., Fernández-Andrés, J., and Sánchez-Soriano, J. (2023). Optimization algorithm to reduce training time for deep learning computer vision algorithms using large image datasets with tiny objects. *IEEE Access* 11, 104593–104605. doi: 10.1109/ACCESS.2023.3316618
- Russo, D. J., Roy, B. V., Kazerouni, A., Osband, I., and Wen, Z. (2018). *A Tutorial on Thompson Sampling, Now Foundations and Trends* (Hanover, Massachusetts, USA: Now Foundations and Trends) 11, 1–96. doi: 10.1561/22000000070
- Ouyang, S., Chen, W., Qin, X., and Yang, J. (2024). Geological background prototype learning-enhanced network for remote-sensing-based engineering geological lithology interpretation in highly vegetated areas. *IEEE J. Selected Topics Appl. Earth Observations Remote Sens.* 17, 8794–8809. doi: 10.1109/JSTARS.2024.3385541

- Salcedo-Sanz, S., Ghamisi, P., Piles, M., Werner, M., Cuadra, L., Moreno-Martínez, A., et al. (2020). Machine learning information fusion in Earth observation: A comprehensive review of methods, applications and data sources. *Inf. Fusion* 63, 256–272. doi: 10.1016/j.inffus.2020.07.004
- Saleem, M. H., Potgieter, J., and Arif, K. M. (2021). *Automation in agriculture by machine and deep learning techniques: A review of recent developments* (Netherlands: Precision Agriculture (Springer)), Vol. 2021. 22, Precision Agriculture. doi: 10.1007/s11119-021-09806-x
- Karmaker, S. K., Hassan, M. M., Smith, M. J., Xu, L., Zhai, C., Veeramachaneni, K., et al. (2021). AutoML to date and beyond: challenges and opportunities. *ACM Comput. Surv* 54, 175.
- Shorten, C., and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *J. big Data* 6, 1–48. doi: 10.1186/s40537-019-0197-0
- Shrestha, A., and Mahmood, A. (2019). “Optimizing deep neural network architecture with enhanced genetic algorithm,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Boca Raton, FL, USA, 2019, 1365–1370. doi: 10.1109/ICMLA.2019.00222
- Srivastava, A., Shinde, T., Joshi, R., Ansari, S. A., and Giri, N. (2021). *Auto-DL: A platform to generate deep learning models* (Singapore: Springer Singapore).
- Sun, Y., Xue, B., Zhang, M., Yen, G. G., and Lv, J. (2020). Automatically designing CNN architectures using the genetic algorithm for image classification. *IEEE Trans. Cybern* 50, 3840–3854. doi: 10.1109/TCYB.6221036
- Towfek, S. K., and Khodadadi, N. (2023). Deep convolutional neural network and metaheuristic optimization for disease detection in plant leaves. *J. Intelligent Syst. Internet Things* 12, 6982. doi: 10.54216/JISIoT.100105
- Yang, W., Yang, C., Hao, Z., Xie, C., and Li, M. (2019). Diagnosis of plant cold damage based on hyperspectral imaging and convolutional neural network. *IEEE Access* 7, 118239–118248. doi: 10.1109/Access.6287639
- Wang, Y., Zhang, Z., Feng, L., Du, Q., and Runge, T. (2020). Combining multi-source data and machine learning approaches to predict winter wheat yield in the conterminous United States. *Remote Sens.* 12, 1232. doi: 10.3390/rs12081232
- Wang, S., Zha, Y., Li, W., Wu, Q., Li, X., Niu, M., et al. (2020). A fully automatic deep learning system for COVID-19 diagnostic and prognostic analysis. *Eur. Respir. J.* 56 (2), 1232. doi: 10.1183/13993003.00775-2020
- Wang, P., Niu, T., Mao, Y., Zhang, Z., Liu, B., and He, D. (2021). Identification of apple leaf diseases by improved deep convolutional neural networks with an attention mechanism. *Front. Plant Sci.* 56, 2000775. doi: 10.3389/fpls.2021.723294
- Watkins, C. J. C. H., and Dayan, P. (1992). Q-learning. *Machine Learning* 8, 279–292. doi: 10.1007/BF00992698
- Wei, D., Chen, J., Luo, T., Long, T., and Wang, H. (2022). Classification of crop pests based on multi-scale feature fusion. *Comput. Electron. Agric.* 194, 106736. doi: 10.1016/j.compag.2022.106736
- Xia, K., Huang, J., and Wang, H. (2020). LSTM-CNN architecture for human activity recognition. *IEEE Access* 8, 56855–56866. doi: 10.1109/Access.6287639
- Xiang, J., Dong, Y., and Yang, Y. (2021). FISTA-net: learning a fast iterative shrinkage thresholding network for inverse problems in imaging. *IEEE Trans. Med. Imaging* 40, 1329–1339. doi: 10.1109/TMI.2021.3054167
- Xiao, X., Yan, M., Basodi, S., Ji, C., and Pan, Y. (2020). Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm. *ArXiv*, 2006–12703. doi: 10.48550/arXiv.2006.12703
- Xu, Y., Hou, S., Wang, X., Li, D., and Lu, L. (2023). A medical image segmentation method based on improved UNet 3+ Network. *Diagnostics* 13, 576. doi: 10.3390/diagnostics13030576
- Yang, S., Zhou, B., Lou, H., Wu, Z., Wang, S., Zhang, Y., et al. (2022). Remote sensing hydrological indication: Responses of hydrological processes to vegetation cover change in mid-latitude mountainous regions. *Sci. Total Environ.* 851, 158170. doi: 10.1016/j.scitotenv.2022.158170
- Yang, L., Yu, X., Zhang, S., Long, H., Zhang, H., Xu, S., et al. (2023). GoogLeNet based on residual network and attention mechanism identification of rice leaf diseases. *Comput. Electron. Agric.* 204, 107543. doi: 10.1016/j.compag.2022.107543
- Yeh, W., Lin, Y. P., Liang, Y. C., and Lai, C. M. (2021). Convolution neural network hyperparameter optimization using simplified swarm optimization. *ArXiv*, 2103.03995. doi: 10.48550/arXiv.2103.03995
- Yu, H., Li, J., Zhang, L., Cao, Y., Yu, X., and Sun, J. (2021). Design of lung nodules segmentation and recognition algorithm based on deep learning. *BMC Bioinf.* 22, 314. doi: 10.1186/s12859-021-04234-0
- Yu, F., Bai, J., Jin, Z., Zhang, H., Yang, J., and Xu, T. (2023). Estimating the rice nitrogen nutrition index based on hyperspectral transform technology. *Front. Plant Sci.* 14, 1118098. doi: 10.3389/fpls.2023.1118098
- Zhao, S., Peng, Y., Liu, J., and Wu, S. (2021). Tomato leaf disease diagnosis based on improved convolution neural network by attention module. *Agriculture* 11, 651. doi: 10.3390/agriculture11070651
- Zheng, W., Xie, H., Chen, Y., Roh, J., and Shin, H. (2022). PIFNet: 3D object detection using joint image and point cloud features for autonomous driving. *Appl. Sci.* 12, 3686. doi: 10.3390/app12073686
- Zimmer, L., Lindauer, M., and Hutter, F. (2021). Auto-pytorch: multi-fidelity metaLearning for efficient and robust autoDL. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 3079–3090. doi: 10.1109/TPAMI.2021.3067763