

# AN ADAPTIVE SUPERPIXELS FOR VEGETATION DETECTION ON HIGH RESOLUTION IMAGES BASED ON MLP

Xiaofang Tang, Xin Huang, Zirou Xiong, Xin Wang\*, Zongqian Zhan

School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China  
(xftang, xhuang1, xzirou)@whu.edu.cn, (zqzhan, xwang)@sgg.whu.edu.cn

Commission III, WG III/1

**KEY WORDS:** Vegetation Detection, Adaptive Superpixels, Multilayer Perceptron (MLP), High-resolution Images

## ABSTRACT:

Vegetation detection aims to find the area which should be attributed with the labels of vegetation on the captured images, such as forest, grass land etc., and nowadays it is a key research topic in the field of remote sensing information processing and application. Over the last few years, the deep learning method based on convolutional neural network (CNN) has become the mainstream method for vegetation detection. However, due to the peculiarities of the underlying encoding and decoding structures, it is common for some CNN methods to lose some boundary details of vegetation when employing high-resolution images with rich details and clear boundaries. In order to improve the boundary localization capability of vegetation, this paper proposes a hybrid solution, i.e., an MLP (MultiLayer Perceptron)-based high-resolution image adaptive superpixels vegetation detection method. Compared with the traditional watershed transform algorithm, this method adopts the two-step boundary marching criterion to generate superpixels with more adherent boundary and compact regularity which contains adaptive neighborhood information by design. Based on the generated superpixels with boundary detail information, this paper applies MLP for binary predictions, i.e., vegetation or non-vegetation. The experimental results show that our method has more precise vegetation boundary localization and higher accuracy compared with several state-of-the-art methods on the UAV image data set and ISPRS data set.

## 1. INTRODUCTION

Vegetation detection is the work of determining the ground vegetation coverage by field inspection, or by investigating remote sensing images or UAV (Unmanned Aerial Vehicle) images. Nowadays, this work plays an important role in the management of agriculture, urban planning for government (Wu et al., 2019; Kwan et al., 2020; Skarlatos et al., 2018). Before the emergence of relevant remote sensing technique, the traditional field inspection which is very costly was often applied to detect vegetation, and results is typically with large subjectivity (Zare et al., 2007). As remote sensing technique develops, one common approach for detecting vegetation is to estimate the normalized difference vegetation index (NDVI) using multispectral or hyperspectral images based on the spectral reflectance properties of vegetation (Bhandari et al., 2012). However, these multispectral or hyperspectral images are expensive to access and their spatial and temporal resolutions are typically limited (Ayhan et al., 2020). In addition, high resolution images are easy to access (such as, UAV platform or consumer cameras) and more flexible for vegetation detection with high precision requirements, showing great potential in vegetation detection.

Recently, learning-based methods, especially deep convolutional neural networks (Huang et al., 2018), have made great progress in pixel-wise land cover classification task. Instead of using hand-crafted features, most CNN-based methods constitute an encode-decode network to learn deep features and predict the label of each pixel (Chen et al., 2017). However, the pooling layer which aims to make CNN architectures perceive large field can lead to the loss of a lot of information, and also the kernel convolution operations ignore the correlation between the global field and the local patches. Furthermore, due to the underlying encode-decode structure, most existing CNN-based methods are not able to recover accurate boundary details between the vegetation and non-vegetation area, which is more serious when dealing with high-resolution images. Although many investigations have been conducted to address this

problem (such as skip connection between encode and decode layers, dilated convolutions) (Yu et al., 2016), it is still an ongoing problem to further improve the boundary details and the corresponding detection results can be consequentially improved as well.

To cope with the abovementioned limitation, in this research we propose a hybrid solution that employs the hand-crafted and learning-based ideas (as Fig.1 illustrates). Specially, considering both the boundary adherence and compactness, an adaptive superpixels generation method for the original image is presented to generate compact superpixels with the details of boundaries kept, the expected superpixels tends to be more feasible for training. Based on each generated superpixels which are regular, the multilayer Perceptron (MLP) is applied in this work for training to do a binary prediction, i.e., vegetation or non-vegetation. The main contributions of this work are as follows:

1. We propose a hybrid solution that employs hand-crafted and learning-based ideas.
2. We propose a method for generating boundary advancing watershed superpixels. In particular, a two-step boundary advancing criterion is suggested to generate superpixels with clearer and more accurate boundaries and more compact regularity.
3. To detect the vegetation, MLP model is trained by integrating of the adaptive neighborhood superpixels.

The rest of this paper is organized as follows. Related works are reviewed in Section 2. The details of our method are illustrated in Section 3. The performance of our experiments and result discussion on superpixels generation and vegetation detection are reported in Section 4. Finally, conclusions and outlooks are drawn in Section 5.

## 2. RELATED WORK

Extensive works have been researched on vegetation detection and

\* Corresponding author.

land cover classification by employing the variants of MLP. In this section, we briefly review some representative traditional super-

pixels algorithm, architectures based on multilayer perceptron, and methods of vegetation extraction relevant to our works.

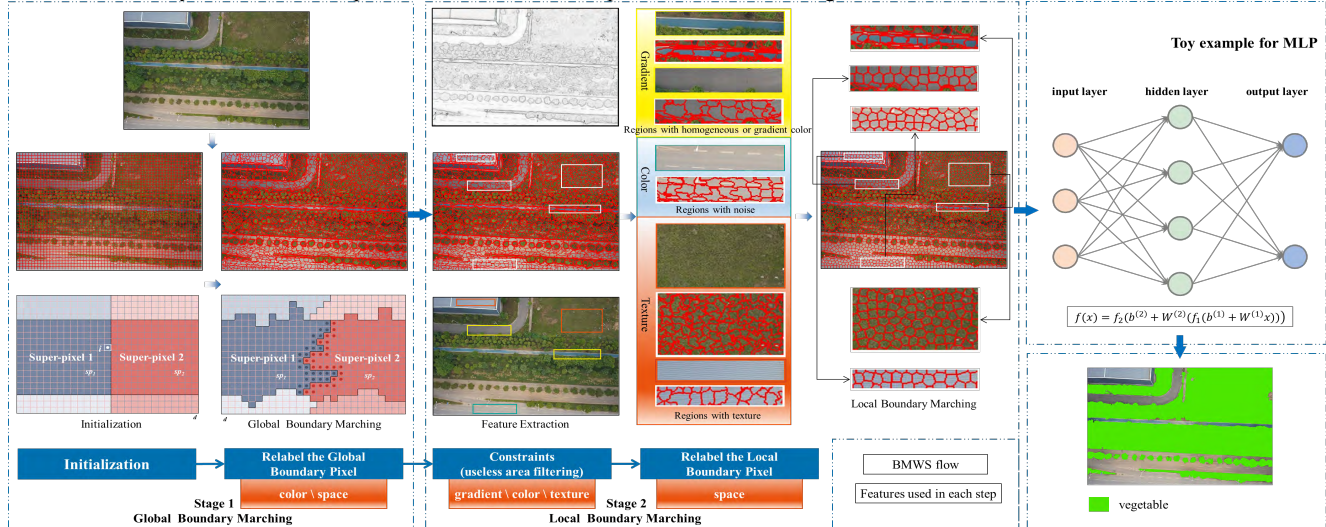
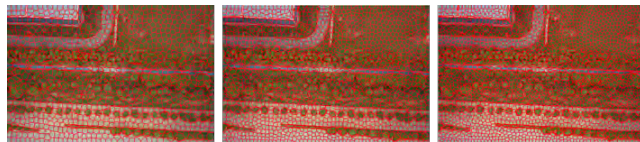


Figure 1. Working flowchart of the proposed hybrid solution for vegetation detection.

## 2.1 Superpixels

To the best of our knowledge, superpixels was originally proposed by Ren et al. (2003). The image is divided into small patches by aggregating adjacent pixels with similar characteristics. These patches are called superpixels. Since then, the application of superpixels has been attracting the attention of researchers from various fields. Superpixels segmentation produces perceptually consistent atomic regions of pixels (Fig. 2). The usage of superpixels instead of original pixels and representing images from pixel-level to region-level (Machairas et al., 2015) can not only allow the original image information to be fully preserved and make the image presentation more intuitive, but also significantly reduce the computational cost of subsequent processing.



(a) BMWS(N=1000) (b) BMWS(N=1500) (c) BMWS(N=2000)

Figure 2. The result of the proposed BMWS algorithm when the number of superpixels is set to 1000/1500/2000.

There are many existing superpixels algorithms, which can mainly be divided into two categories: deep learning-based algorithms and traditional algorithms. Deep learning methods usually utilize costly hardwares to train the corresponding neural networks. Whereas, traditional superpixels algorithms are easier to implement (Chaibou et al., 2017), which also has two main categories: watershed-based and clustering-based (Zhang et al., 2021; Xiao et al., 2018). Some representative traditional methods are shown in Tab. 1. It can be seen that the time efficiency of watershed superpixels is generally higher. This section briefly reviews several representative superpixels methods.

**(1) Watershed (W):** In an image of size  $R \times L$  with  $N$  superpixels, the pixel priority  $P_w$  of watershed (Beucher et al., 1992) is:

$$P_w = D_{gra} \quad (1)$$

A smaller value  $P_w$  indicates a higher priority. The original gradient labeling watershed algorithm is efficient and fast, but suffers from the problems of segmentation irregularity and the compactness is sensitive to the free parameters.

**(2) Compact Watershed (CW):** Based on watershed-based segmentation, a spatial constraint was introduced (Neubert et al., 2014), and the modified priority criterion can be summarized as:

$$P_{cw} = D_{gra} + \lambda_s \cdot D_{spa} \quad (2)$$

Where  $D_{gra}$  is the gradient distance between a pixel and its adjacent pixels,  $D_{spa}$  is the spatial distance between a pixel and the corresponding seed, and  $\lambda_s$  is the compactness parameter which is related to the number of superpixels and image size, this method is with high computational efficiency, but low boundary localization capability.

**(3) Spatial Constrained Watershed (SCoW):** SCoW considers gradient information when adding spatial constraints on the basis of the original watershed (Hu Z et al., 2015). The priority  $P_{SCoW}$  is defined as:

$$P_{SCoW} = g_{RGB} + \lambda_s \cdot e^{-g_{RGB}} \cdot D_{spa} \quad (3)$$

SCoW uses RGB color gradients, and  $\lambda_s$  is the compactness parameter, this method can generate more regular superpixels, but its boundary accuracy is low.

**(4) Watershed-based Superpixels with Global and Local boundary marching (WSGL):** WSGL takes the sum of HSV color homogeneity denoted by  $D_{color}$  and size constraints denoted by  $D_{size}$  as the priority  $P_{WSBM}$ , and then performs global boundary marching and local boundary marching (Yuan et al., 2020).

$$P_{WSBM} = D_{color} + \lambda_s \cdot D_{size} \quad (4)$$

However, only gradient features are considered, which results in that superpixels from rich-texture regions are still very irregular despite no clear object boundaries. Furthermore, this method starts with the pre-selected seeds, followed by boundary refinement, which is complicated.

**(5) Simple Linear Iterative Clustering (SLIC):** SLIC (Achanta et al., 2012) is the most classic superpixels algorithm based on clustering, which applies local k-means clustering and iteratively refines the clustering results until convergence. The distance between pixels is defined as:

$$D = \sqrt{D_{col}^2 + \lambda_s \cdot D_{spa}^2} \quad (5)$$

where  $D_{col}$  and  $D_{spa}$  are the color distance and spatial distance respectively, and  $\lambda_s$  is a parameter that is related to the compactness. SLIC is currently one of the most widely used superpixels algorithms, its accuracy and compactness can achieve an acceptable balance by tuning  $\lambda_s$ , but clustering takes a long time.

| Method | Type             | time  | Year | Color | Gradient | Space | Texture | Smoothness | Size | Number | Iteration |
|--------|------------------|-------|------|-------|----------|-------|---------|------------|------|--------|-----------|
| W      | Watershed-based  | 0.13s | 1993 | --    | ✓        | --    | --      | --         | --   | --     | --        |
| CW     | Watershed-based  | 0.16s | 2014 | --    | ✓        | ✓     | --      | ✓          | --   | ✓      | --        |
| SCoW   | Watershed-based  | 0.17s | 2015 | ✓     | ✓        | ✓     | --      | ✓          | --   | ✓      | --        |
| WSGL   | Watershed-based  | 0.25s | 2020 | ✓     | ✓        | ✓     | --      | ✓          | ✓    | ✓      | --        |
| MS     | Clustering-based | 25.9s | 2002 | --    | --       | ✓     | --      | --         | --   | --     | ✓         |
| SLIC   | Clustering-based | 0.37s | 2012 | ✓     | --       | ✓     | --      | ✓          | --   | ✓      | ✓         |
| LSC    | Clustering-based | 1.09s | 2015 | ✓     | --       | ✓     | --      | ✓          | --   | ✓      | ✓         |
| CAS    | Clustering-based | 1.01s | 2018 | ✓     | ✓        | ✓     | ✓       | --         | --   | ✓      | ✓         |

**Table 1.** Superpixels algorithms.

## 2.2 Multilayer Perceptron

Rosenblatt et al. (1958) designed the first neural network (perceptron), they used a computer to simulate how the human brain worked, which in fact is a forward network with a layer of neurons activated by threshold activation functions. Werbos et al. (1981) introduced the multilayer perceptron (MLP) in the neural network back-propagation (BP) algorithm, and it became the key technology of neural network architecture and significantly contributed to supervised-learning methods.

MLP is a multilayer feedforward neural network with at least three layers (Fig. 1): an input layer that receives signals, an output layer that provides predictions of results, and many hidden layers consisting of neurons with nonlinear activation functions. The number of hidden layers and neurons in MLP may vary up to the tasks, the more complicated the task is, the more neurons are required in general, while overfitting needs to be taken care of on a large MLP (Murtagh et al., 1991).

Multi-layer perceptron has many advantages (Liu et al., 2021), such as high stability, easy for parallel processing and etc. Therefore, MLP has been popularly applied in classification and vegetation prediction tasks (Ayhan et al., 2020), Tang et al. (2022) used residual learning to improve the feature extraction ability of MLP network, and 98.48% precision was obtained in remote sensing image classification. Jahani et al. (2020) introduced the MLP model to improve the feature detection ability of vegetation changes, which is successful to predict changes in vegetation density. Raczko et al. (2018) used a feed-forward multilayer perceptron (MLP) with a single hidden layer to identify tree species and forests, achieving an average overall accuracy of 87%. Xie et al., (2019) used SLIC superpixels segmentation and trained MLP model for urban green land classification, proving that the MLP method based on ultra-high spatial resolution imagery is suitable for urban green land monitoring.

## 2.3 Vegetation Detection

Many related algorithms have been applied to detect vegetation on remote sensing imagery, such as, K nearest neighbor algorithm (KNN) (Xu et al., 2013), clustering algorithm (Yang et al., 2020), support vector machine algorithm (SVM) (Sun et al., 2012), wavelet-based classification method (Huang et al., 2006) and all connected neural networks (Li et al., 2019) and semi-supervised algorithms (Yang et al., 2019), but they still have limitations in training speed and detection precision (Jia et al., 2013). The training of KNN is not time-efficient, SVM is applicable for large-scale data.

In general, satellite remote sensing images is often employed to detect vegetation. In the past few years, thanks to the development

of learning-based CNN methods, many CNN architectures are taken over for detection task, such as ResNet (He et al., 2016) and HRNet (Sun et al., 2019). The discriminability of the features extracted from these networks is gradually improving, and one remaining problem is the loss of detail in vegetation detection due to inter-pixel pooling operations and convolution operations during encoding (Yang et al., 2020a). Chen et al. (2019) used deep learning models such as GoogleNet to classify and detect vegetation areas, and achieve the precision of 88.3%. In addition, Huang et al. (2016) applied the adaptive resonance network to vegetation detection, and obtained a precision rate of 90.8%. However, their model is relatively complex with a large number of parameters. Wang et al. (2019) detected vegetation information in forest areas based on the traditional convolutional neural network (CNN), and their precision is around 94.0%. However, scenes of forest areas are relatively simple, and this method has limitation on identifying the detailed information of vegetation boundaries.

In order to guarantee the precision of vegetation detection and solve the problem of boundary details on high resolution images, this paper introduces adaptive superpixels with accurate boundary positioning and compact regularity, and then a corresponding multilayer perceptron is trained to obtain better vegetation detection results.

## 3. METHODOLOGY

The main goal of this work is to make full use of the high-resolution information by superpixels which consider increased heterogeneity in the same region and homogeneity in different regions, and by training a MLP model to detect vegetation on high resolution images with good boundary details. Therefore, there are two parts, namely, superpixels generation and MLP which are corresponds to the hand-crafted and learning-based ideas. In the following, more details will be introduced regarding these two parts.

### 3.1 Boundary Marching Watershed Superpixels (BMWS)

The boundary marching watershed superpixels algorithm proposed in this section contains only one free parameter for the number of superpixels, which can be easily adapted to images with different resolutions.

(1) According to the original image content, the images are suggested to be classified into meaningful and meaningless regions. In general, meaningful regions contain more object boundaries (for example, the contour boundary of houses, trees and roads as is shown in Fig. 1). Conversely, meaningless regions are typically homogeneous regions with identical gradients (e.g., bicycle lanes and roads without sign lines), noisy regions (e.g., roads with sign lines), and regions with densely and rich textures (e.g., grass and the top of houses).

(2) Two different criteria that works on the meaningful and meaningless regions are investigated, respectively. One is responsible for taking care of the boundary adherence in the meaningful regions to improve the segmentation accuracy; the other one is to filter the local meaningless regions information by investigating several features with respect to gradients, color and texture. Compactness and regularity of the desired superpixels are constrained by above features. According to these two criteria, the whole superpixels generation procedure contains two stages: firstly, the global boundary marching. After initialization, the constraints of color and space consistency are considered to re-label the boundary pixels and improve the superpixels from initialization. Secondly, we use the constraints of gradient, color and texture to filter out local meaningless regions, and only space features are used on the filtered regions to re-label boundary pixels for improving compactness. These two steps are adopted for the global boundary adherence and the compactness of the superpixels on the local meaningless regions, respectively, which can result in superpixels with strong boundary adherence, regular and compact shape on high-resolution images. The following sections describe each feature and how it is calculated in detail.

**3.1.1 Feature representation:** Image  $I$  with size  $P = R \times L$  (Row, Column) is represented by  $i = (i_1, i_2, \dots, i_P)$ . By manually setting the initial number of superpixels  $N_{\text{manual}}$ , we get  $N$  actual generated superpixels represented by  $sp = (sp_1, sp_2, \dots, sp_N)$ , each pixel can be represented by a feature vector as  $i_p = [l_p, a_p, b_p, x_p, y_p, g_p, t_p]^T$ ,  $p = 1, 2, \dots, P$ , and this formula can also be used to describe superpixels or edges. Among them,  $[l, a, b]^T$  represent color features,  $[x, y]^T$  represent spatial attribute features, whereas  $g$  and  $t$  denote gradient features and texture features, respectively.

**(1) Color features:** It is defined as,

$$D_{\text{color}}(p, q) = \sqrt{c_1(l_p - l_q)^2 + c_2(a_p - a_q)^2 + c_3(b_p - b_q)^2} \quad (6)$$

$$c_1 + c_2 + c_3 = 3 \quad (7)$$

$$c_2 = c_3 = 0.5 \times (3 - c_1) \quad (8)$$

where  $D_{\text{color}}$  represents the color distance of the two elements  $p$  and  $q$ , which could be pixels, superpixels or edges;  $l$ ,  $a$  and  $b$  represent the brightness and two color dimensions in the CIELAB color space (Xiao et al., 2018);  $c_1$ ,  $c_2$  and  $c_3$  are their respective weights,  $c_1 = 0.1$ .

**(2) Spatial feature:** In this paper, the Euclidean distance between elements is used to represent spatial features to enhance the compactness of superpixels,

$$D_{\text{space}}(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2} \quad (9)$$

$D_{\text{space}}(p, q)$  is the Euclidean distance of  $p$  and  $q$ ,  $x$  and  $y$  are the horizontal and vertical coordinates, respectively.

**(3) Gradient feature:** To estimate the gradient, the Sobel operator which can preserve clear detail is employed in this work, and the corresponding gradient feature Gradient ( $I$ ) is defined as,

$$\text{Gradient}(I) = \sqrt{G_x^2 + G_y^2} \quad (10)$$

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (11)$$

where,  $I$  represents the image,  $G_x$  and  $G_y$  represent the gray value of the detected horizontal and vertical edges, respectively.

**(4) Texture features:** The WLD (Weber Local Descriptor) (Xiao et al., 2018) is used to reflect the intensity relationship between the central pixel and its neighboring pixels in a fixed window, which is defined as,

$$t_c = \arctan\left(\frac{\sum_{i=1}^8 x_i - x_c}{x_c + k}\right) \quad (12)$$

where  $t_c$  is the texture feature of the center pixel  $c$ ,  $x_c$  is the intensity value of  $c$ ,  $x_i$  is the intensity value of each adjacent pixel of  $c$ , and  $k$  is a small constant value that prevents the denominator from being zero. The texture distance between elements  $p$  and  $q$  is defined as,

$$D_{\text{texture}}(p, q) = |t_p - t_q| \quad (13)$$

If  $p$  or  $q$  is a superpixels or edge, the texture value is the average texture feature of all included pixels.

**3.1.2 Global Boundary Marching:** To ensure both the performance of segmentation and the adherence as much as possible, a criterion is established using color features and spatial feature. First, all boundary pixels are marked according to Equations 14. The process repeats until all adjacent boundary pixels are relabeled, or stops until the next queue reaches the M1 (as equation (16) computed) queue, the marking criterion and the side length  $d$  of the initialized partition superpixels are established as follows:

$$L'(i) = \begin{cases} L(sp_1), D_{\text{color}}(sp_1, i) + \lambda D_{\text{space}}(sp_1, i) \leq \\ D_{\text{color}}(sp_2, i) + \lambda D_{\text{space}}(sp_2, i) \\ L(sp_2), D_{\text{color}}(sp_1, i) + \lambda D_{\text{space}}(sp_1, i) > \\ D_{\text{color}}(sp_2, i) + \lambda D_{\text{space}}(sp_2, i) \end{cases} \quad (14)$$

$$d = \sqrt{\frac{R \times L}{N_{\text{manual}}}} \quad (15)$$

$$M_1 = \max\left(\frac{d}{2}, 10\right) \quad (16)$$

where,  $sp_1$  and  $sp_2$  are two adjacent superpixels sharing a boundary, pixels on the boundary  $i \in sp_1$ ,  $L'(i)$  is the new label of the pixel.  $L(sp_1)$  and  $L(sp_2)$  are the labels of  $sp_1$  and  $sp_2$ , respectively, and  $\lambda$  is the compactness parameter,  $D_{\text{color}}(sp_1, i)$  and  $D_{\text{space}}(sp_1, i)$  do not include boundary pixel  $i$ .

**3.1.3 Local Boundary Marching:** We design to filter four types of content meaningless regions based on gradient, color and texture constraints.

The gradient feature  $\Gamma_g$  with respect to the boundary  $\varepsilon_1$  is defined as:

$$\Gamma_g(\varepsilon_1) = \begin{cases} 1, & g_{\varepsilon_1} \leq \eta_g \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

where  $\eta_g$  is the gradient threshold, which is relabeled when  $\Gamma_g = 1$  for the boundary  $\varepsilon_1$ .

The color constraint  $\Gamma_c$  on the boundary  $\varepsilon_1$  is defined as:

$$\Gamma_c(\varepsilon_1) = \begin{cases} 1, & D_{\text{color}}(\varepsilon_1, sp_2) \leq \eta_c \text{ and } g_{\varepsilon_1} \leq 3\eta_g \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

where  $\eta_c$  is the gradient threshold, which is relabeled when  $\Gamma_c = 1$  for the boundary  $\varepsilon_1$ .

The color constraint  $\Gamma_t$  on the boundary  $\varepsilon_1$  is defined as:

$$\Gamma_t(\varepsilon_1) = \begin{cases} 1, & D_{\text{texture}}(sp_1, sp_2) < \eta_t \text{ and } \varpi_t(sp_1) \geq \eta_{\varpi} \\ & \text{and } D_{\text{color}}(sp_1, sp_2) \leq \eta_c \text{ and } g_{\varepsilon_1} \leq 4\eta_g \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

$$\varpi_t(sp) = \sqrt{\frac{\sum_{i=1}^{n_{sp}} (t_i - u_{sp})^2}{n_{sp}}} \quad (20)$$

where  $\varpi_t$  represents the texture variance, which is used to indicate that the color and gradient feature is in fact due to texture information (not to two homogeneous superpixels).  $n_{sp}$  is the number of pixels in the superpixel  $sp$ ,  $u_{sp}$  is the average texture

feature ( $D_{texture}$ ) value of  $sp$ ,  $\eta_t$  and  $\eta_{\omega}$  are texture and variance thresholds, respectively, which are relabeled when  $\Gamma_t = 1$  for boundary  $\varepsilon_1$ .

When any of these three constraints is 1, edges are identified for possible relabeling using a compactness-enhancing criterion as follows:

$$\Gamma(\varepsilon_1) = \begin{cases} 1, \Gamma_s(\varepsilon_1) = 1 \text{ or } \Gamma_c(\varepsilon_1) = 1 \text{ or } \Gamma_t(\varepsilon_1) = 1 \\ 0, \text{ otherwise} \end{cases} \quad (21)$$

$$L''(i) = \begin{cases} L(sp_1), \Gamma(\varepsilon_1) = 1 \text{ and } \Gamma(\varepsilon_2) = 1 \text{ and} \\ D_{space}(sp_1, i) > D_{space}(sp_2, i) \\ L(sp_2), \text{ otherwise} \end{cases} \quad (22)$$

$$M_2 = \frac{d}{2} \quad (23)$$

where,  $\Gamma = 1$  indicates that the corresponding edge can be relabeled,  $L''(i)$  is the new label of pixel  $i$  in the local boundary marching, and pixels on the boundary  $i \in sp_1$ . The boundary pixels between the superpixels ( $sp_1$  and  $sp_2$ ) can be relabeled only when both the boundaries  $\varepsilon_1$  and  $\varepsilon_2$  satisfy the constraints, and only spatial features are involved in the relabeling criteria.

The relabeling process of local boundary marching is also implemented by priority queue. Unlike global boundary marching, in this step, only the pixels that satisfy the conditions (pixels with  $\Gamma = 1$ ) are pushed to the front queue. The process can be repeatedly popped and marked accordingly until all boundary pixels are marked or the M2 queue is reached. This stage allows to maximize the compactness of all superpixels in meaningless regions.

### 3.2 Multilayer Perceptron, MLP

After generating the superpixels, the next goal is to determine which superpixels are vegetation. One intuitive solution is to apply multilayer perceptron. As Fig. 1 shows, in this work, the MLP is trained to do a binary prediction based on the generated superpixels. Although the compactness is taken into consideration in previous step, there are superpixels with various sizes, in addition, some irregular shapes from the superpixels still exist which is not suitable to be cast as an input for MLP. To ease this limitation, average features of all pixels within the superpixels are computed as the input.

MLP includes input layer, hidden layer, and output layer, which is a typical fully connected network. The complete MLP model process is divided into two processes: forward propagation and back propagation.

The forward propagation calculates the output of the neuron through the pre-trained MLP model. The feature vector of the input layer is the superpixels feature  $x_{ij}$ , and the weights and biases of the layers are represented by  $w_{ij}$  and  $b_{ij}$  respectively,  $f$  represents the activation function, and the output value  $a_j$  of the hidden layer node is defined as follows:

$$a_j = f\left(\sum_{i=1}^m w_{ij} x_{ij} + b_{ij}\right) \quad (24)$$

$m$  is the number of features of the input superpixels. Taking the output value of the hidden layer as the input of the output layer, the final result expression of the output layer is:

$$y_k = f(a_k) = f\left(\sum_{j=1}^n w_{kj} x_{ij} + b_{kj}\right) \quad (25)$$

$n$  is the number of hidden layers, and  $k$  represents the  $k$ th output.

Back propagation is used to train weights and biases of the model, the total error is first computed by using weights  $w$  and bias  $b$  of current iteration, and then adjust the weights and biases to minimize the total error. The error function  $E$ , also known as the loss function, is defined as:

$$E = \frac{1}{2} \sum_{k=1}^n (y_k - t_k)^2 \quad (26)$$

$t_k$  is the ground truth label,  $y_k$  is the predicted label, and the loss is iteratively refined by using the gradient descent method and the weights and biases are recursively updated as following equation:

$$\begin{cases} w_{ij} = w_{ij} - \eta \frac{\partial E(w, b)}{\partial w_{ij}} \\ b_{ij} = b_{ij} - \eta \frac{\partial E(w, b)}{\partial b_{ij}} \end{cases} \quad (27)$$

where,  $\eta$  is the learning rate, in the range (0, 1].

## 4. EXPERIMENTS

To demonstrate the performance of the suggested method, the ISPRS benchmark and some self-generated UAV images are tested, in which the main goal of the designed experiments is to evaluate our vegetation detection result as a whole. In particular, the generated superpixels will be firstly assessed, and then the subsequent vegetation detection results will be discussed. In the following sections, we will introduce the dataset, experimental settings and experimental results in detail.

### 4.1 Dataset and implementation details

**4.1.1 UAV Image Dataset:** This dataset constitutes optical images captured by UAVs, including 600 images with a size of 7952×5304 pixels, and the spatial resolution is about 0.016m. We manually create labels containing background, cars, buildings, roads (footpaths), patches of grass and individual rows of trees, and labels of grass and trees are marked as vegetation and all the remaining categories are as non-vegetation.

**4.1.2 ISPRS Vaihingen Challenge Dataset:** This is an ISPRS 2D Semantic Labeling Challenge benchmark dataset (Wegner et al., 2017), including 33 remote sensing images (top) in NIR, red, and green bands with a spatial resolution of 0.09m. Only the top image is used in our experiments. The labels of the dataset contain six categories, namely, impervious surfaces, low vegetation, trees, cars, buildings, and background miscellaneous. In this section, the low vegetation and tree labels in the ground-truth are regarded as vegetation, and the rest of the categories are regarded as non-vegetation.

**4.1.3 Experimental settings:** All experiments (except ResNet and HRNet) were performed on a machine with six 3.20GHz Intel(R) Core (TM) i7-8700 processors and 12 threads available, 8GB memory. The MLP activation function is linear rectification function (Relu function), which can effectively overcome the problem of gradient disappearance. The maximum number of iterations of pre-training is set to 10, the weight decreasing parameter (to prevent overfitting) is set to 3e-3, and the sparse penalty term (Supplementary adjustment of the loss function) is set to 3, the sparse parameter is set to 0.1. The maximum number of iterations of fine-tuning is set to 100, and the weight decreasing parameter is set to 1e-4. The number of channels in the two hidden layers is 128 and 64, respectively. The two datasets are trained with 10,000 superpixels respectively, and the trained SP-MLP model is obtained to do the superpixels vegetation detection.

**4.1.4 Evaluation metrics:** The performance of models on different datasets is assessed by precision, recall, F1-score, overall accuracy (OA) and Intersection-over-Union (IoU). The formulas of precision, recall, F1-score, Accuracy and IoU are expressed as follows:

$$precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN} \quad (28)$$

$$F_1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (29)$$



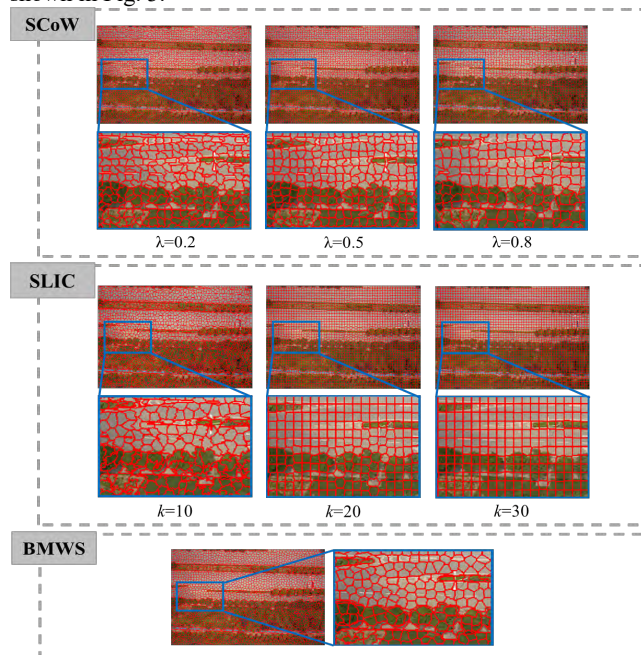
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (30)$$

$$IoU = \frac{TP}{TP + FP + FN} \quad (31)$$

where  $TP$ ,  $FP$ ,  $TN$ ,  $FN$  denote the true positive, false positive, true negative and false negative pixels.

## 4.2 Comparison of Superpixels

**4.2.1 Compactness and adherence analysis of BMWS:** BMWS is a superpixels algorithm that is supposed to be with good performance of both compactness and adherence. To demonstrate this, several state-of-the-art methods including the classic SLIC (Achanta et al., 2012) and SCoW (Hu Z et al., 2015) algorithms are compared with, the number of superpixels is set to 1500. Examples of UAV images superpixel results from the SLIC and SCoW algorithms with different compactness parameters are shown in Fig. 3.



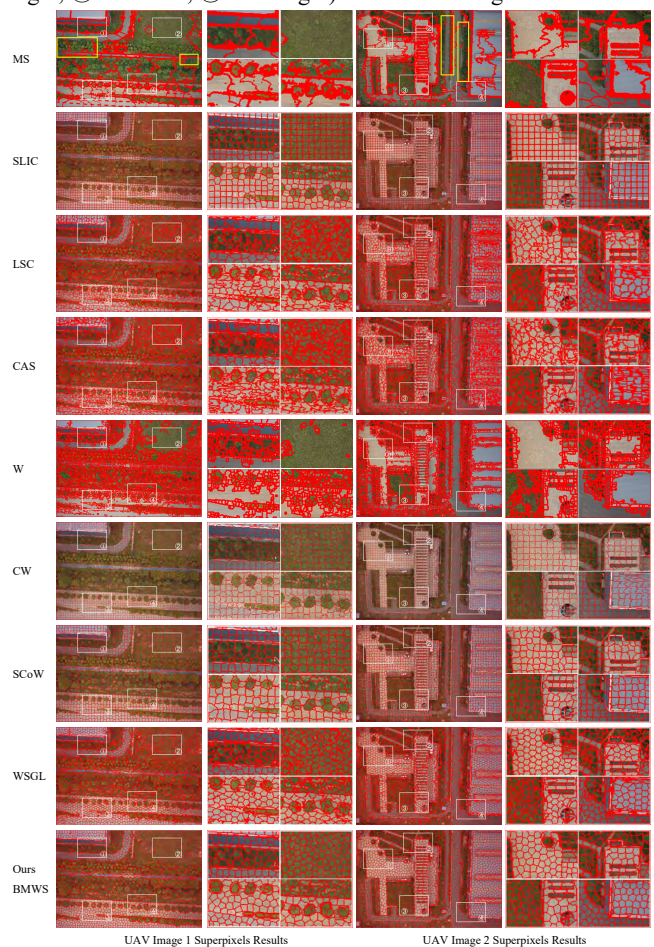
**Figure 3.** Comparison of compactness and adherence of superpixels algorithm. The parameters  $\lambda$  and  $k$  are used to adjust the compactness (regularity) of superpixels.

From Fig. 3, the result of SCoW is not compact, and the SLIC result is too regular with poor boundary adherence. On the contrary, in the area with rich vegetation, our method can not only ensure the compactness of the meaningless regions, but also shows superior adherence to the boundary, which to some extent solves the defect that the adherence and compactness of most superpixels algorithms are with mutual inhibition.

**4.2.2 Extensive comparison with state-of-the-art superpixels algorithms:** In this section, the clustering-based MS (Comaniciu et al., 2002), SLIC, LSC (Chen et al., 2017), CAS (Xiao et al., 2018) and watershed-based W, CW, SCoW, WSGL algorithms are used to compare with the BMWS algorithm suggested this paper, and two UAV images are tested for qualitative evaluation.

UAV image 1 is used to demonstrate the algorithm's ability to adhere to borders and compactness on grassland, and UAV image 2 is used to demonstrate the effectiveness of our method on dealing with texture-rich cases. All parameters were manually adjusted for the best results following the default settings or the suggestion of the original paper. The number of MS and W superpixels cannot be fixed, and the number of other superpixels is set to 1500. The

segmentation results and their local details (①upper left, ②upper right, ③lower left, ④lower right) are shown in Fig. 4.



**Figure 4.** Extensive comparison experiments of Superpixels.

As shown in Fig. 4 and Fig. 5, the shape of the superpixels segmentation result obtained by the BMWS algorithm is more regular in the meaningless area, and the corresponding boundaries looks better. MS shows that the generated boundary of a target is not preserved any more (in the yellow rectangle) which is the worst result, and the time efficiency is not desirable as well (the average time for processing one UAV and ISPRS image is 2697.36s and 1415.35s, respectively). And, W cannot provide compact superpixels (the corresponding average time of W on each UAV and ISPRS image is 16.22s and 3.89s, respectively). The superpixels of SLIC is typically with regular polygon, however the pixels of different targets are fused in the same superpixels, which will reduce the subsequent segmentation accuracy and the boundary information cannot be well kept. The two algorithms of LSC and CAS are able to attach to the boundary, but the deformation of superpixels blocks is serious, and some extremely deformed superpixels are generated, moreover, the superpixels shape is still irregular even in homogeneous areas, it is difficult to visually distinguish the details of various objects, and the corresponding computational complexity is relatively high. The CW and SCoW can generate the superpixels in a very fast way, but shows weak boundary adherence and irregular shapes, especially in the area with low gradient changes. The WSGL algorithm only considers gradient features, and in regions which is full of rich textures and have no object boundaries, superpixels are still very irregular. The BMWS algorithm are shown to be able to have strong adherence at meaningful areas, and in the meaningless areas a relatively high compactness can be obtained. In Fig. 5, our method is in general the fastest or second fastest solution for yielding superpixels.

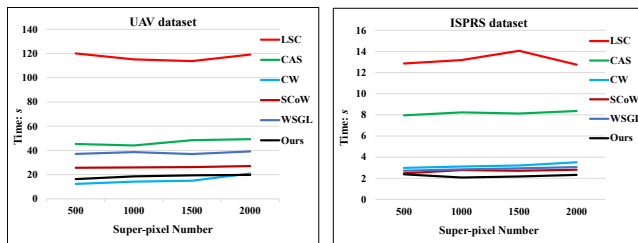


Figure 5. Superpixels runtime (unit: s)

To sum up, comparing with other superpixels methods, our BMWS method can generate the best or the second best superpixels with least or the second least running time.

### 4.3 Comparison of vegetation detection

Several state-of-the-art methods are compared on the UAV dataset and the ISPRS Vaihingen dataset, namely, ResNet (He et al., 2016) and HRNet (Sun et al., 2019), pixel-based MLP models, SLIC-MLP and SCoW-MLP models that takes the generated superpixels of SLIC and SCoW as the input of MLP, and the proposed SP-MLP. Among them, the number of SLIC, SCoW and BMWS superpixels for each image is set to 2000.

ResNet and HRNet are trained with the SGD optimizer on NVIDIA GTX 3090ti GPUs. The base learning rate are set to 0.001. A poly learning rate policy is employed, in which the initial learning rate is multiplied by  $(1 - \frac{epoch}{total\_epoch})^{1.0}$  during each epoch. The momentum value is 0.9 and the weight decay value is  $1e-4$ . For each experiment, the training procedure is with 100 epochs and validation is applied every 5 epochs. The results of the two datasets are described in detail next.

**4.3.1 UAV dataset experiment results:** The vegetation prediction results of the UAV dataset are shown in Fig. 6. We replace the vegetation directly on the image, and the green area is the vegetation detection result.

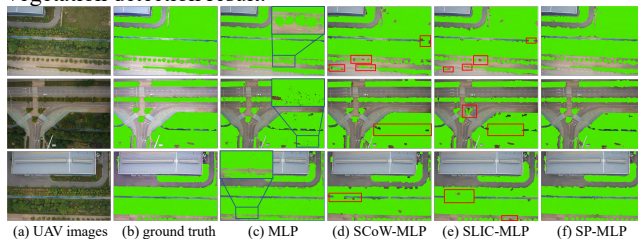


Figure 6. UAV image vegetation detection map.

According to the results and the corresponding details shown in Fig. 6, it can be found that in some areas with rich vegetation but simple features, the pixel-based MLP vegetation extraction results are noisy and there are some misclassified areas. The detection results after SCoW and SLIC superpixels segmentation can alleviate the noise problem to some extent. Qualitatively, the vegetation detection results of the SP-MLP algorithm give a more complete vegetation segmentation.

|          | precision     | recall        | F1-score      | OA            | IoU           |
|----------|---------------|---------------|---------------|---------------|---------------|
| MLP      | 94.18%        | 95.92%        | 95.04%        | 94.49%        | 90.55%        |
| SCoW-MLP | 95.66%        | 93.86%        | 94.74%        | 94.16%        | 90.02%        |
| SLIC-MLP | 95.69%        | 94.47%        | 95.08%        | 94.92%        | 90.82%        |
| proposed | <b>96.86%</b> | <b>95.22%</b> | <b>96.03%</b> | <b>95.93%</b> | <b>92.94%</b> |

Table 2. Evaluation of vegetation detection on UAV images dataset. Best results are highlighted in bold font.

Tab. 2 provides a quantitative evaluation, compared with the pixel-

based MLP detection results, the F1-score, OA, and IoU of the SP-MLP are superior by 0.9%, 1.4%, and 2.3%, respectively. Compared with the results of SCoW and SLIC superpixels segmentation, the IoU of our method is improved by 2.1% and 2.9%. In conclusion, we find that better superpixels results will improve the performance of vegetation detection to a certain extent, while poor superpixels will lead to some erroneous detections and reduce the precision.

**4.3.2 ISPRS dataset experiment results:** Fig. 7(a) and (b) shows the three original ISPRS images and the ground truth vegetation areas. The vegetation detection results of various methods are shown in Fig. 7(c)-(h). In general, the pure deep learning-based semantic segmentation models ResNet and HRNet can yield very promising results (based on large training dataset, we crop training areas into images of  $512 \times 512$  pixel size and obtained 4326 training images and 111 validation images through augmentation methods such as rotation.). By investigating the ground truth and the original images, we found that the provided ground truth contains few incorrect labels (such as, the black rectangular box and its local details are enlarged), and these two CNN-based method shows basically identical result with ground truth. At the same time, their performance to keep boundaries is in fact worth further improvement promising. Our proposed method, on the other hand, it is more prone to keep boundaries, and rarely misidentifies non-vegetation as vegetation (even sometimes when the training true labels from the ground truth are incorrect).

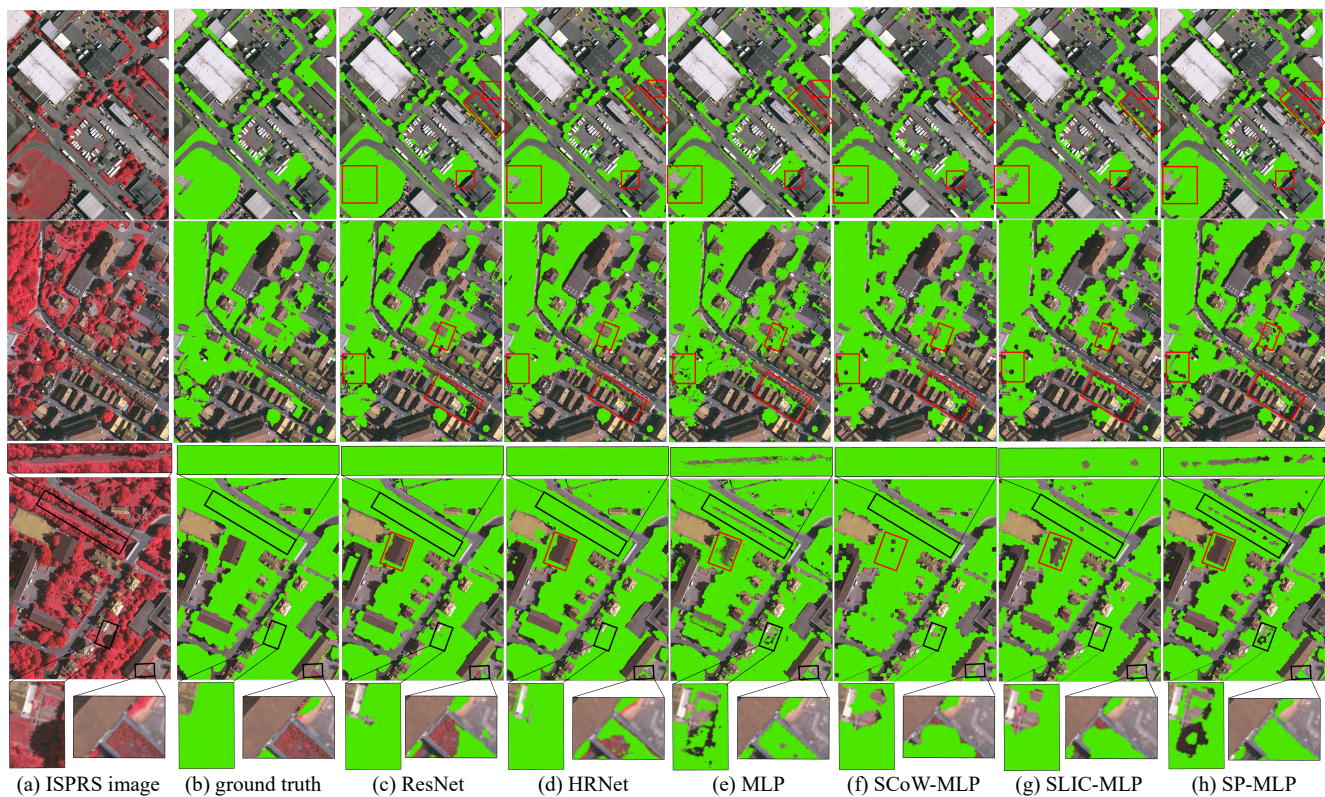
In the pixel-based MLP results (Fig.7(e)), the vegetation detection is incomplete, the boundaries are not clear, and explicit salt and pepper errors emerge. Then, SLIC and SCoW fail to capture some details (Fig.7(f), (g)), which result in some small single tree undetected, and there are a large number of false prediction. Looking at the results in the red rectangle, in general, our method can preserve most boundary details, this can be explained by the fact that our proposed superpixels algorithm is designed to provide more suitable adaptive neighborhood information.

|          | precision     | recall        | F1-score      | OA            | IoU           |
|----------|---------------|---------------|---------------|---------------|---------------|
| ResNet   | 95.33%        | <b>95.51%</b> | <b>95.42%</b> | <u>96.29%</u> | <u>91.25%</u> |
| HRNet    | <u>95.51%</u> | 93.47%        | 94.47%        | 95.62%        | 89.53%        |
| MLP      | 95.14%        | 91.81%        | 93.43%        | 94.59%        | 87.68%        |
| SLIC-MLP | 92.51%        | 90.93%        | 91.61%        | 93.22%        | 84.55%        |
| SCoW-MLP | 88.42%        | 91.58%        | 89.90%        | 91.72%        | 81.70%        |
| SP-MLP   | <b>96.07%</b> | <u>94.60%</u> | <u>95.33%</u> | <b>96.55%</b> | <b>91.27%</b> |

Table 3. Vegetation detection and evaluation results of ISPRS image dataset. The best results are bolded and the second-best results are underlined.

The quantitative evaluation results of the ISPRS dataset are shown in Tab. 3. Investigating the result of MLP and our method, the efficacy of the our superpixels are demonstrated, as MLP using the original pixel for training are inferior regarding all evaluation metrics. Next, comparing with the methods that integrate with superpixels (SLIC-MLP and SCoW-MLP), the proposed method is much superior in every evaluation metric (precision, recall, F1-score, OA and IoU), this better performance can be attributed to the proposed superpixels generation method which can provide more adherent, compact and regular superpixels. As for the pure deep learning methods (ResNet and HRNet), our vegetation detection result is basically better than HRNet, and for ResNet, comparable performance can be achieved by our method, but, the prediction model of ours is much smaller and much easier to train.





**Figure 7.** Three ISPRS original images and vegetation extraction results

## 5. CONCLUSION

We propose a hybrid solution for vegetation detection for high resolution images based on an adaptive superpixels and MLP, in which hybrid means hand-crafted and learning-based ideas are employed. To generate superior superpixels, a two-step boundary marching criterion is suggested to ensure more adherent boundary and compact regularity. Based on the generated superpixels with boundary detail information, a MLP is trained for binary predictions. The experimental results show that on the UAV dataset, vegetation detection result of our method can achieve 96.86% precision, 95.22% recall, 96.03% F1-score, 95.93% OA and 92.92% IoU. On the ISPRS Vaihingen Challenge dataset, compared with the existing methods, i.e., ResNet, HRNet, our method has achieved comparable results, and the boundary details of our method are visually better.

Overall, our SP-MLP method can perform well on UAV dataset and the ISPRS Vaihingen dataset, but several issues may need further consideration. Up to now, only one target (vegetation) are investigated in the proposed method, there is a possibility to extend our work on multiple targets detection. Moreover, to improve the performance of MLP, more reasonable features from superpixels of various shapes are worth studying.

## ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 61871295).

## REFERENCES

- Achanta, R., Shaji, A., Smith, K., et al. 2012. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis & Machine Intelligence*. vol. 34, no. 11, pp. 2274-2282.
- Ayhan, B., Kwan, C., Budavari, B., Kwan, L., Lu, Y., Perez, D., Li, J., Skarlatos, D., Vlachos, M., 2020. Vegetation Detection Using Deep Learning and Conventional Methods. *Remote Sensing*. 12(15):2502.
- Beucher, S., Meyer, F., 1993. The Morphological Approach to Segmentation: The Watershed Transformation. In *Mathematical morphology in image processing*. CRC Press. pp. 433-481.
- Bhandari, A.K., Kumar, A., Singh, G.K. 2012. Feature extraction using Normalized Difference Vegetation Index(NDVI): A case study of Jabalpur city. *Procedia Technol.* 6, 612–621.
- Chaibou, M.S., Conze, P.H., Kalti, K., et al. 2017. Adaptive strategy for superpixel-based region-growing image segmentation. *Journal of Electronic Imaging*. 26(6): 061605.
- Chen, B., Wang, H.Z., Xu, X. L. et al. 2019. The classification by medium resolution remote sensing images based on deep learning algorithm of GoogleNet model. *Bulletin of Surveying and Mapping*. 2019(6): 29-33, 40.
- Chen, J., Li, Z., Huang, B., 2017. Linear Spectral Clustering Superpixel. *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3317-3330.
- Chen, L. C., Papandreou, G., Kokkinos, K., Murphy, K., Yuille, A.L., 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4), 834-848.
- Comaniciu, D., Meer, P., 2002. Mean shift: a robust approach toward feature space analysis. *IEEE Trans Pattern Analysis & Machine Intelligence*. 24(5):603-619.
- He K., Zhang X., Ren S., Sun, J., 2016. Deep residual learning for image recognition. *IEEE conference on computer vision and pattern recognition*, pp. 770-778.
- Hu, Z., Qin, Z., Li, Q., 2015. Watershed Superpixel. *IEEE International Conference on Image Processing (ICIP)*. IEEE. pp. 349-353.



- Huang, B., Zhao, B., Song, Y., 2018. Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery. *Remote Sensing of Environment*. 214, 73–86
- Huang, C.D., Xu, F., 2016. Extraction of sparse vegetation based on true color high resolution remote sensing image. *Journal of Geomatics*, 41(5): 42-46, 50.
- Huang, X., Zhang, L.P., LI, P. X., 2006. Methods for classification of the high spatial resolution remotely sensed images based on wavelet transform. In *Geomatics and Information Science of Wuhan University*. Vol. 31, No. 12, pp. 1055-1058.
- Jahani, A., Goshtasb, H., Saffariha, M., 2020. Tourism impact assessment modeling of vegetation density for protected areas using data mining techniques. *Land Degradation & Development*. 31(12): 1502-1519.
- Jia, K., Yao, Y. J., Wei, X. Q. et al., 2013. A review on fractional vegetation cover estimation using remote sensing. *Advances in earth science*, 28(7): 774-782.
- Kwan, C., Gribben, D., Ayhan, B., et al. 2020. Improving Land Cover Classification Using Extended Multi-Attribute Profiles (EMAP) Enhanced Color, Near Infrared, and LiDAR Data. *Remote Sensing*. 12(9):1392.
- Li, Z. H., 2019. Rough image classification based on BP neural network. *Public CommunSciTechnol*, 11(19):78-79.
- Liu, R., Li, Y., Liang, D., et al. 2021. Are we ready for a new paradigm shift? A Survey on Visual Deep MLP. *arXiv preprint arXiv:2111.04060*.
- Machairas, V., Faessel, M., Cardenas-Pena, D., et al. 2015. Waterpixels. *IEEE Transactions on Image Processing*. 24(11):3707-3716.
- Murtagh, F., 1991. Review of Y. Pao's Adaptive Pattern Recognition and Neural Networks, in *Journal of Classification*, 8, 115–119.
- Neubert, P., Protzel, P., 2014. Compact Watershed and Preemptive SLIC: On Improving Trade-offs of Superpixel Segmentation Algorithms. 2014 22nd International Conference on Pattern Recognition (ICPR). *IEEE Computer Society*. pp. 996-1001.
- Raczko, E., Zagajewski, B., 2018. Tree species classification of the UNESCO man and the biosphere karkonoski national park (poland) using artificial neural networks and APEX hyperspectral images. *Remote Sensing*. 10(7): 1111.
- Ren, X., Malik, J., 2003. Learning a classification model for segmentation. *Proceedings Ninth IEEE International Conference on Computer Vision*. vol.1, pp. 10-17.
- Rosenblatt, F., 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review* 65(6):386-408.
- Skarlatos, D., Vlachos, M., 2018. Vegetation removal from UAV derived DSMS, using combination of RGB and NIR imagery. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* IV-2, 255–262.
- Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., et al., 2019. High-Resolution Representations for Labeling Pixels and Regions. *arXiv preprint arXiv: 1904.04514*.
- Sun, Y. B., 2012. Texture image classification algorithm based on support vector machine. *ComputSimul*, 29(5) :287-290.
- Tang, X., Liu, X., Yan, P., et al. 2022. An MLP network based on residual learning for rice hyperspectral data classification. *IEEE Geoscience and Remote Sensing Letters*. vol. 19, pp. 1-5.
- Wang, J. 2019. Application of deep learning algorithm in mountain vegetation classification. *Value Eng*, 38(4): 161-163.
- Wegner, J.D., Rottensteiner, F., Gerke, M., Sohn, Gunho, 2017. The ISPRS labelling challenge. Available in the WWW:<http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html>.
- Werbos, P.J., 1981. Applications of advances in nonlinear sensitivity analysis. In *Proceedings of the 10th IFIP Conference*,. 31.8 - 4.9, NYC, pages 762–770.
- Wu, Z., Xu, X., Tao, S., 2019. Analysis of Urban Green Space Extraction Based on UAV Images. *Journal of Sichuan Forestry Science and Technology*. 40(06):65-70.
- Xiao, X., Zhou, Y., Gong, Y.J., 2018. Content-Adaptive Superpixel Segmentation. *IEEE Transactions on Image Processing*. 27(6) : 2883-2896.
- Xie, G., Niculescu, S., Lam, C., et al. 2019. Machine learning methods and classification of vegetation in Brest, France. *Remote Sensing Technologies and Applications in Urban Environments IV*. *International Society for Optics and Photonics*. 11157: 111570J.
- Xu, X.Y., 2013. Study on K-nearest neighbor algorithm oriented to weed identification. Harbin : Northeast Agricultural University.
- Yang, C., Rottensteiner, F., Heipke, C., 2020a. Investigations on Skip-Connections with an Additional Cosine Similarity Loss for Land Cover Classification. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3, pp. 339-346.
- Yang, H.X., Yang, X.B., Kou, Z. Y., et al. 2019. K plane clustering algorithm based on L1 norm. *JNanjing Univ Aeronaut Astronaut*, 51(5): 681-686.
- Yang, H.X., Yang, X.B., Zhang, F.Q. et al. 2020. Semi-supervised plane clustering Algorithm. *JNanjing Univ (NatSci)*, 56(1): 9-18.
- Yang, X.B., Ge, Y.Q., Zhang, F.Q. et al., 2019. Semi-supervised algorithm for forest fire recognition based on matrix pattern. *JGraph*, 40(5): 835-842.
- Yu, F., Koltun, V., 2016. Multi-Scale Context Aggregation by Dilated Convolutions. *International Conference of Neural Networks on Computer Vision and Pattern Recognition*, *arXiv preprint arXiv: 1511.07122*.
- Yuan, Y., Zhu, Z., Yu, H., et al. 2020. Watershed-Based Superpixels With Global and Local Boundary Marching. *IEEE Transactions on Image Processing*, 29: 7375-7388.
- Zare, A., Bolton, J., Gader, P., Schatten, M., 2007. Vegetation mapping for landmine detection using long-wave hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* 46(1), 172–178.
- Zhang, Y., Liu, M., He, J., et al. 2021. Affinity fusion graph-based framework for natural image segmentation. *IEEE Transactions on Multimedia*. vol. 24, pp. 440-450.