Front. Struct. Civ. Eng. 2022, 16(4): 414–433
https://doi.org/10.1007/s11709-021-0797-6

RESEARCH ARTICLE

Optimal CNN-based semantic segmentation model of cutting slope images

Mansheng LIN[a], Shuai TENG[a], Gongfa CHEN[a*], Jianbing LV[a], Zhongyu HAO[b]

[a] *School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou 510006, China*
[b] *JSTI Group Guangdong Inspection and Certification Co. Ltd, Nanjing 210000, China*
[*] *Corresponding author. E-mail: gongfa.chen@gdut.edu.cn*

ABSTRACT This paper utilizes three popular semantic segmentation networks, specifically DeepLab v3+, fully convolutional network (FCN), and U-Net to qualitively analyze and identify the key components of cutting slope images in complex scenes and achieve rapid image-based slope detection. The elements of cutting slope images are divided into 7 categories. In order to determine the best algorithm for pixel level classification of cutting slope images, the networks are compared from three aspects: a) different neural networks, b) different feature extractors, and c) 2 different optimization algorithms. It is found that DeepLab v3+ with Resnet18 and Sgdm performs best, FCN 32s with Sgdm takes the second, and U-Net with Adam ranks third. This paper also analyzes the segmentation strategies of the three networks in terms of feature map visualization. Results show that the contour generated by DeepLab v3+ (combined with Resnet18 and Sgdm) is closest to the ground truth, while the resulting contour of U-Net (combined with Adam) is closest to the input images.

KEYWORDS slope damage, image recognition, semantic segmentation, feature map, visualizations

1 Introduction

The stability of slopes plays a vital role in the construction and operation of highway systems [1]. In recent years, a large number of high slopes have been established in China due to the rapid development of highway construction. The slopes in certain mountainous areas may be more prone to or are already suffering from damages caused by natural environment and human activity, making it crucial to evaluate the current state of important sections of the slopes to prevent further accidents and losses.

At present, there are various methods to assess the slope stability. For example, most engineers measure slope stability using sensor data and mathematical calculations. Wu et al. [2] designed a Portrait-based Disaster Alerting System using hillslope monitoring sensors, which combines service servers, wireless sensor networks, and analytic network processing technology to predict and monitor slope disasters. Another way to assess slope stability is by observing the overall appearance, which heavily relies on the inspectors to regularly assess the slope, obtain relevant digital images, and make an intuitive judgment on possible damages or risks through comparison and experience [3]. Slope inspections may be impeded due to height differences between slopes or rocky road conditions, thus Unmanned Aerial Vehicles (UAVs) have been introduced to collect slope images [4]. In general, UAVs are used to collect images of lattice beams, retaining walls, and vegetation on cutting slopes. Surface damages can be identified by comparing slope images collected from different time periods. However, manually classifying and identifying damage is time-consuming, inefficient, and subjective due to the large number of collected images [5], making an artificial intelligence method imperative for slope damage detection.

In recent years, artificial intelligence is mostly used to monitorslopestabilitythroughslopedisplacement(thedisplacement change of preset observation points), similar to

Article history: Received Oct 5, 2021; Accepted Nov 14, 2021

using a nonlinear function to establish the mapping between the input and output (e.g., the bending analysis of Kirchhoff plate [6], boundary value problems [7], and solution of partial differential equations in computational mechanics [8]). Zhou et al. [9] used the dynamic energy factor, slope factor, and resistance factor as the input of a back propagation (BP) neural network and predicted the maximum vertical and horizontal displacement during a slope collapse, with an accuracy of 86.67% and 93.33%, respectively. Xing et al. [10] used the genetic algorithm to optimize the parameters of a support vector machine (SVM) and established a slope stability prediction model. Lin et al. [11] trained an artificial neural network (ANN) with 955 highway slope samples to study the influence of earthquakes on slope failure characteristics. However, these methods require intensive instrumentation when it comes to large-scale civil infrastructures (e.g., slopes), including an enormous amount of sensor installation and data collection [12]. Thus, using sensor results as the input of artificial intelligence assessment in slope stability is still a challenge.

As opposed to image recognition of slope surfaces where data collection is much easier than slope displacement monitoring, Wu [4] used SVM and convolution neural network (CNN) to classify two types of slopes: landslide and no-disaster. However, methods such as ANN and SVM are limited by their low detection accuracy, overfitting phenomenon, and slow detection speed [13]. At the same time, artificial intelligence in the field of image detection requires a large amount of training image samples to improve model accuracy. Therefore, a rapid, automatic feature extraction algorithm is necessary to process the immense slope monitoring data [14,15]. Hence, deep convolution neural networks (DCNNs) have been increasingly utilized for slope image detection for their stronger robustness and lower computational cost [16]. Ghorbanzadeh et al. [17] used a CNN method to detect slope failure. Spectral information and slope data were derived from the detected topographic data, specifically UAV remote sensor images. Shu et al. [3] compared the performance of two networks, AlexNet and GoogleNet, to classify slope disaster images and results showed that GoogleNet can reach an accuracy of approximately 90%. The above studies show that the category and location of slope damages can be obtained through image classification and location. However, these methods can only estimate the rough position and contour of objects of interest in the images, thus a more accurate method is needed to improve image recognition for slope disaster prevention.

The deep learning method has been gradually applied for extracting object contours, making pixel level classification methods imperative. These methods can directly classify image pixels to identify objects of interest. There are many semantic segmentation CNN models based on pixel level classification, such as SegNet [18], U-Net [19], DeepLab v3+ [20], and fully convolutional network (FCN) [21], where the network performance and segmentation results are different. In civil engineering, Narazaki et al. [22] used SegNet and FCN to segment bridge structural components in images of complex scenes. The sequential configuration method based on an FCN achieves 100% score of mean intersection of union (*MIoU*). Liu et al. [23] used the You Only Look Once (YOLO) and a modified U-Net to extract the contour of pavement cracks on a sidewalk with a precision of 97.24%. Dung and Anh [24] used an FCN to automatically extract crack images with an average precision of 90%.

Another artificial intelligence method, automated vision-based inspection, has attracted much attention in civil engineering and widely used in the fields of modal strain energy and vibration responses of a steel truss [25] for road crack detection [23], bridge component detection [22], and structural damage feature extraction. However, improving image recognition is crucial to establish the mapping relationship between the degree of damage of key slope components and its future stability to prevent possible slope disasters. Since the majority of existing guidelines of structural inspections rely on both damage and structural information to evaluate overall structural stability, timely detection of key components and slope displacement is crucial [26]. Thus, applying semantic segmentation to the field of image recognition would be an ideal solution.

Different semantic segmentation networks use different feature extraction and expansion strategies, thus using the appropriate models can improve the recognition efficiency of cutting slope components. This paper compares and tests the performance of three semantic segmentation networks (DeepLab v3+, FCN, and U-Net) in seven categories (lattice beam, vegetation, vegetation disappearance, retaining wall, sky, road and road sign). In addition, different optimizers are used to determine the ideal combination for cutting slope image recognition. Finally, the segmentation results of different networks are analyzed through feature map visualization.

2 Methodology

This section introduces the principle, procedure, and three examples of semantic segmentation.

2.1 Semantic segmentation

In this paper, three semantic segmentation networks, DeepLab v3+, FCN, and U-Net, are used to identify

cutting slope images (Fig. 1). A semantic segmentation network usually consists of two parts: a contracting path (feature extraction) and an expansive path (feature expansion).

The contracting path extracts features from the input image and creates a feature map. It can be generally regarded as an ordinary classification neural network without the classification layer [27]. In Fig. 2, the contour of a lattice beam in the input image is gradually extracted by the contracting path, which will then connect with the expansive path. Some models may have feature extraction integrated in the expansive path [19,20].

In the expansive path, an enlargement of the feature map size is completed by transposed convolution layers [28], whose main function is to restore the feature map of the contracting path into the same resolution of the input image (Fig. 2). The classification operation is completed by a 1 × 1 convolution layer to output the same number of feature maps as the class number (i.e., 7) defined by the network. The element value of the feature map represents the "strength" in which a certain position belongs to a certain class.

The feature maps will then pass through the Softmax layer to transform its relative "strengths" into the probability. The probability distribution matrices with the same size (length and height) as the input image are obtained. The values of each matrix represent the probability of their corresponding position category in the input image. For example, if the probability of category A is the largest, the final layer of the network will set the label in this pixel as A.

This paper uses weighted cross-entropy as the loss function, which functions to quickly narrow the gap between the predicted and real value of the network model through its own gradient direction and size, similar to minimizing the cost function (loss function) to predict the potential energy in physics-informed neural networks (PINNs) [29].

For the cross-entropy loss function, the loss of a batch of training samples is defined as:

$$loss_{\text{batch1}} = -\frac{1}{bv}\sum_{n=1}^{b}\sum_{m=1}^{v}\sum_{l=1}^{u} w(l)\cdot y_{nml}\cdot \log\hat{y}_{nml}, \quad (1)$$

where b is the amount of training samples in a batch, v is the number of pixels of an image, u is the class number, $w(l)$ is the weight, y_{nml} is the probability of real label, and $\hat{y}_{nml}$ is the probability of prediction label. The loss function is calculated for each batch.

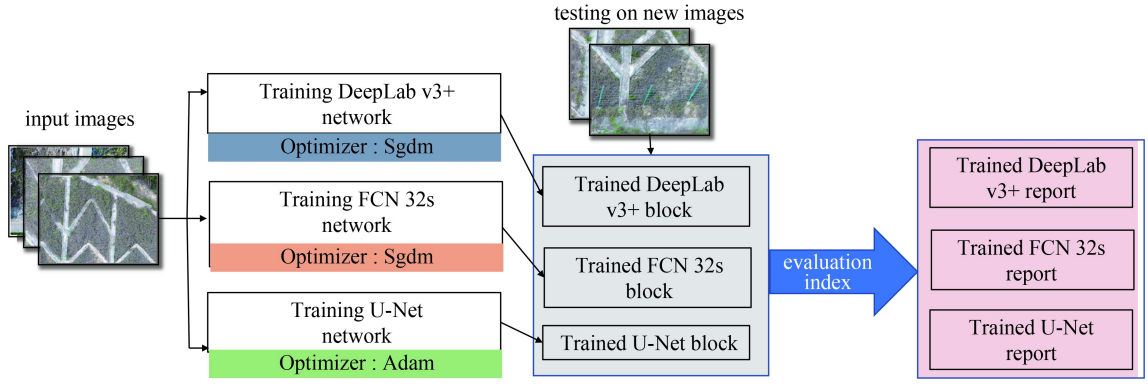


Fig. 1 Flow chart of cutting slope image segmentation.

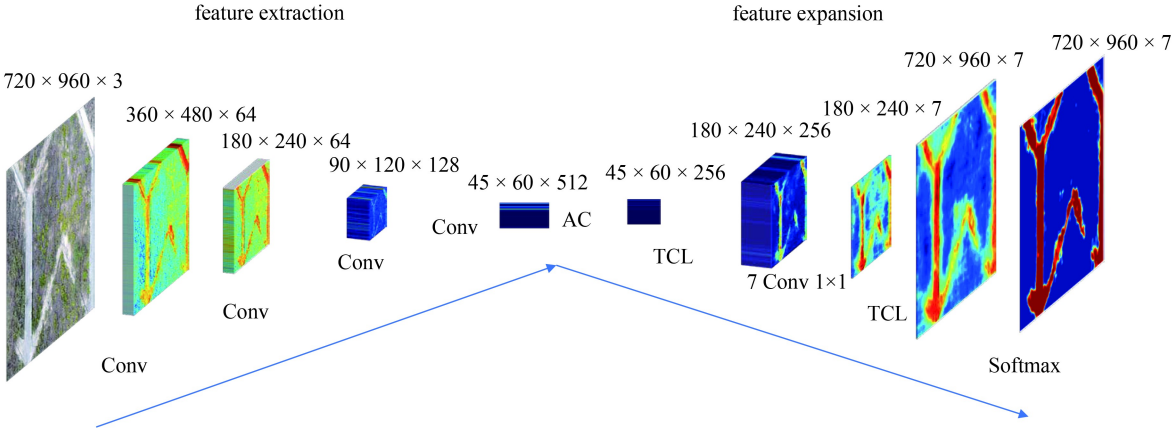


Fig. 2 Extraction and expansion process of a DeepLab v3+ with Resnet18 model. AC = atrous convolution, TCL = transposed convolution layer.

In short, feature extraction of the semantic segmentation neural network transforms the input image into a multidimensional feature representation (similar to CNN's automatic feature extraction from input data [30]), while the expansion path is a shape generator that produces object segmentation from the feature extracted from the convolution network. The final output of the network is a probability distribution matrix with the same size as the input image. The category of each pixel is determined according to the probability distribution matrix [27].

2.1.1 DeepLab v3+

Section 2.1.1 introduces the semantic segmentation network, DeepLab v3+, whose feature extraction path uses Resnet18 [31]. Prior to feature expansion DeepLab v3+ performed four different strategies of atrous convolution (AC) [32] to roughly extract the features of the previous-layer feature maps (Table 1). The purpose of padding is to center the contour of the object of interest to increase the probability of it being extracted by the AC layer to extract it [33].

There will also be "skip" operations in the semantic segmentation network when the shallow feature map generates two branch paths. One of the branch paths is continuously extracted by a convolution kernel and rectified linear unit (Relu) layer [34], and the other is a "deep concatenation" of shallow feature maps and deep feature maps (Fig. 3) [19].

In addition, the feature extraction layer in DeepLab v3+ uses batch normalization, whose purpose is to scale the input to the nonlinear activation function at each layer using the learned mean and standard deviation parameters, thus accelerating and improving the convergence of the parameter updating process [35]. Figure 4 shows the network structure of DeepLab v3+ with the Resnet18 model, where L1–L67 represent the Resnet18 network structure without the output layer.

2.1.2 Fully convolutional network

The feature extraction path of the FCN is based on the VGG16 network. The network structure of the FCN expansion path will differ due to the size of upsampling factors (8, 16, and 32). The main difference between them is the feature expansion. The FCN upsampling factor used in this paper is 32 (represented by FCN 32s), meaning only one transposed convolution is used [21].

FCN 32s (Fig. 5) directly uses the transposed convolution layers to expand and resize the feature image into the size of the input image, while the DeepLab v3+ network

Table 1 Properties of AC layer

property	1st AC	2nd AC	3rd AC	4th AC
padding size	0	6	12	18
dilation factor (DF)	1	6	12	18
old filler size (OFZ)	1×1	3×3	3×3	3×3
new filler size (NFZ)	1×1	11×11	23×23	35×35
new convolution kernel				

Note: The difference between the four ACs is: a) the padding size; b) the dilation factor (DF of the vertical and horizontal directions are the same). NFZ = (OFZ−1) × DF−1, the rest of the positions are filled with 0. The old filler becomes the new filler (AC) through different DF indicators.

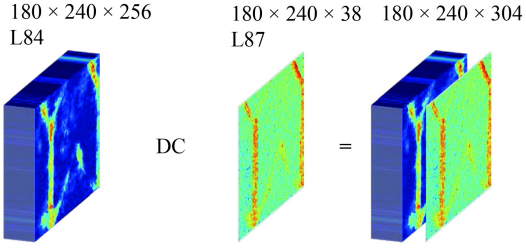


Fig. 3 Deep concatenation (DC). L84 (feature maps after multiple feature extraction) and L87 (feature maps directly from shallow "skip") deep concatenation.

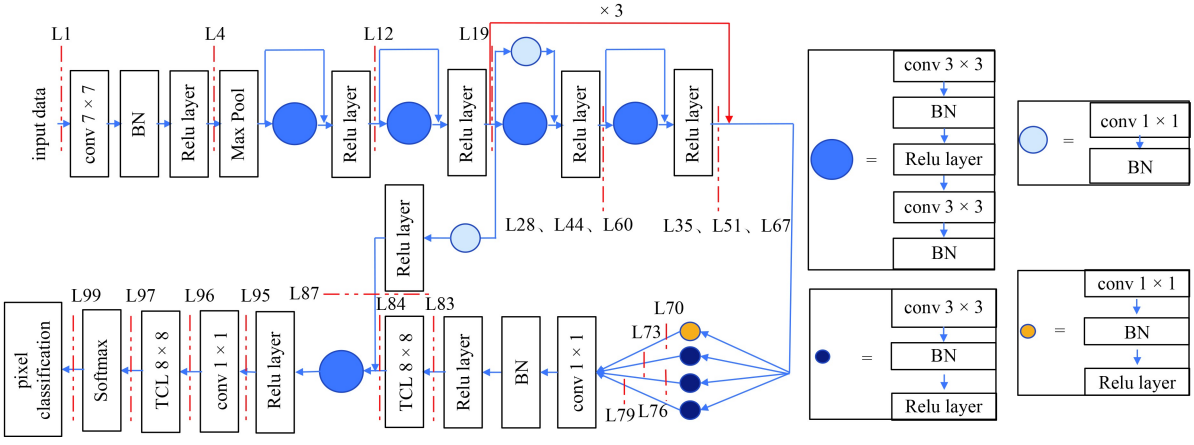


Fig. 4 DeepLab v3+ with Resnet18 network structure, where L*i* represents the *i*th layer in the network, *i* = 1,...,99. BN = Batch Normalization; TCL = transposed convolution layer.

will continue feature extraction even after the transposed convolution layers are used.

In addition, dropout layers are used in FCN. Dropout [36] prunes the outputs of designated layers during training to improve the robustness of the trained network, reducing the overfitting effect.

2.1.3 U-Net

Section 2.1.3 introduces the U-Net network, which was given its name due to its U-shaped structure. Its feature extraction path consists of convolution layers, Relu layers, and Max Pool layers, while the feature expansion path is comprised of transposed convolution layers and Relu layers.

Compared with the previous semantic segmentation networks, the U-Net structure (Fig. 6) has many "deep concatenation" operations that regularly connect various shallow feature maps with deep feature maps.

To conclude, the three semantic segmentation networks utilize different feature extraction and expansion strategies. For DeepLab v3+ (combined with resnet18), L2–L83 are the feature extraction parts, while L84–L99 are the feature expansion parts. For FCN 32s, L2–L39 are the feature extraction parts, while L40–L42 are the feature expansion parts. For U-Net, L2–L27 are the feature

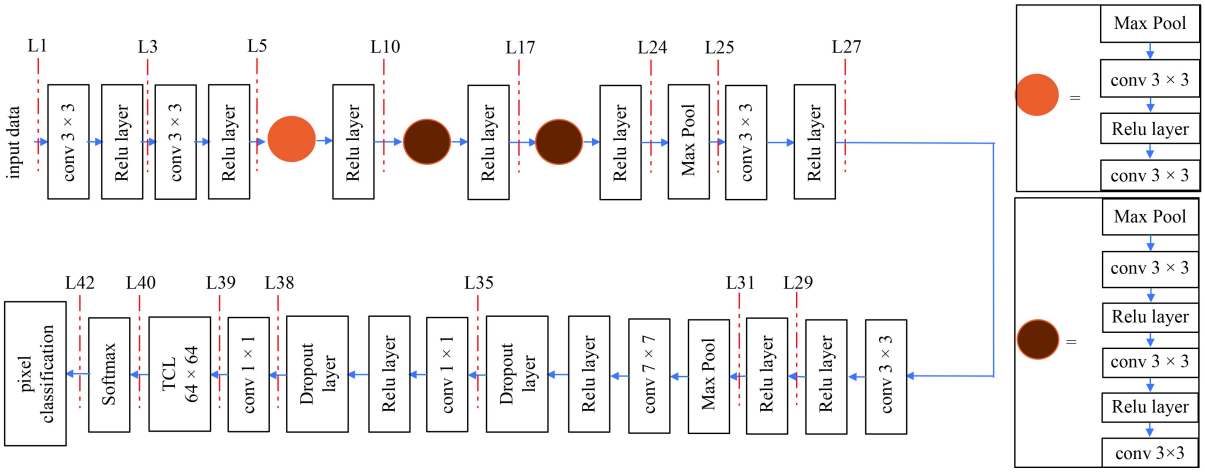


Fig. 5 FCN 32s network structure.

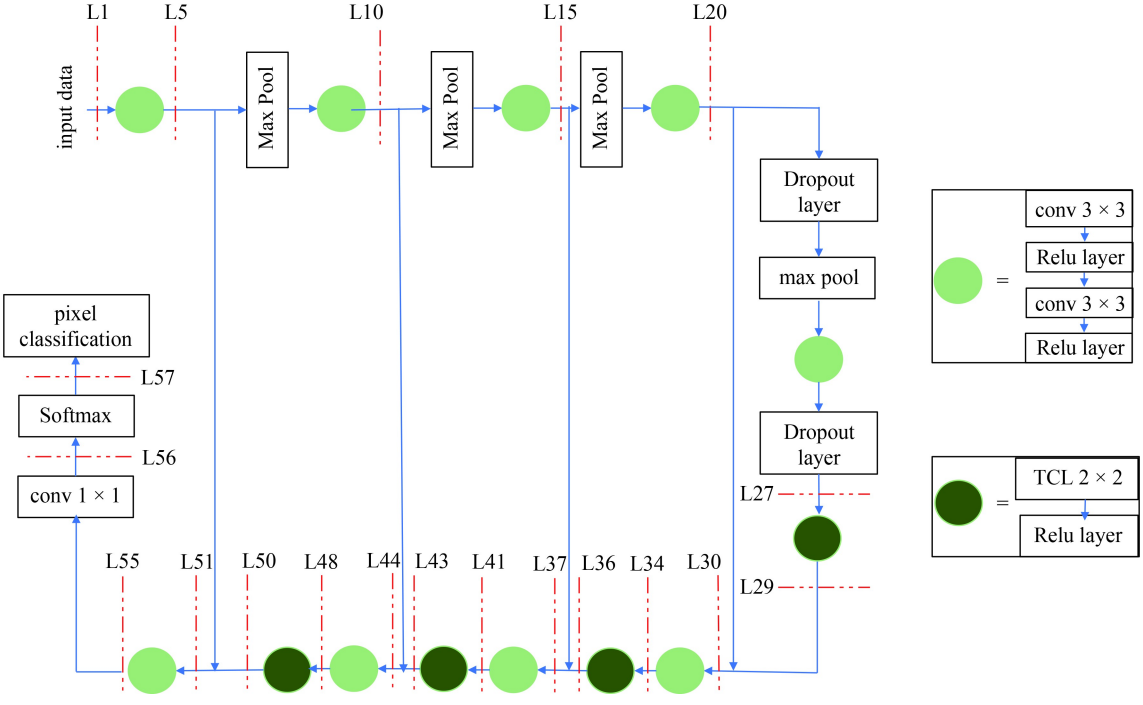


Fig. 6 U-Net structure.

extraction parts, while L28–L57 are the feature expansion parts. The biggest difference between these is the length of feature expansion since the number of deep concatenation is different (DeepLab v3+ is 1, FCN 32s is 0, U-Net is 4). The following sections introduce the indicators for evaluating the performance of these networks, which play a decisive role in selecting the optimal cutting slope semantic segmentation network model.

2.2 Evaluating indices

This paper uses the following evaluation indices: precision (*PR*), per-class pixel accuracy (*PA*), intersection of union (*IoU*), mean boundary F1 score of a class (*MBFSoC*), mean pixel accuracy (*MPA*), global pixel accuracy (*GPA*), *MIoU*, weighted intersection of union (*WIoU*), and mean boundary F1 score of a dataset (*MBFSoDS*) [37]. It should be noted that, in the semantic segmentation task, true positive (*TP*), false positive (*FP*), false negative (*FN*), and true negative (*TN*) are all based on image pixels. In the following formulas, *NC* and *P* represent the category number and image number, respectively, while p_{ij} represents the probability that the pixel with a real label i is predicted as with a label j.

$$\begin{aligned} TP_i &= \sum p_{ii}, \\ FP_i &= \sum p_{ij}(i \neq j), \\ FN_i &= \sum p_{ji}(i \neq j), \\ TN_i &= \sum p_{ij} - TP_i - FP_i - FN_i, \end{aligned} \tag{2}$$

$$PR_i = TP_i/(TP_i + FP_i), \tag{3}$$

$$PA_i = TP_i/(TP_i + FN_i), \tag{4}$$

$$\begin{aligned} MBFSoC_i = &\left(\sum_{i=1}^{NP}(2 \times PR_{i,\,\text{picture}} \times PA_{i,\,\text{picture}}/\right. \\ &\left.(PR_{i,\,\text{picture}} + PA_{i,\,\text{picture}}))\right)/NP, \end{aligned} \tag{5}$$

$$MPA = \left(\sum_{i=1}^{NC} PA_i\right)/NA, \tag{6}$$

$$GPA = \left(\sum_{i=1}^{NC} TP_i\right)/\sum p_{ij}, \tag{7}$$

$$MIoU = \left(\sum_{i=1}^{NC} IoU_i\right)/NC, \tag{8}$$

$$WIoU = \left(\sum_{i=1}^{NC}((TP_i + FN_i) \times IoU_i)\right)/\sum p_{ij}, \tag{9}$$

$$MBFSoDS = \left(\sum_{i=1}^{NC} MBFSoC_i\right)/NC. \tag{10}$$

In addition to the above evaluation indices, this paper also uses the false positive rate (*FPR*), mean false positive rate (*MFPR*), and global false positive rate (*GFPR*) to evaluate network prediction error, which is calculated as followed.

$$FPR_i = FP_i/(TN_i + FP_i), \tag{11}$$

$$MFPR_i = \left(\sum_{i=1}^{NC} FPR_i\right)/NC, \tag{12}$$

$$GFPR_i = \left(\sum_{i=1}^{NC} FP_i\right)/\sum p_{ij}. \tag{13}$$

2.3 Establishment of dataset

This subsection explains the source and image processing method of the dataset, as well as the setup of the training parameters.

2.3.1 Dataset processing

In order to train the pixel classifiers of slope components, it is necessary to collect relevant training samples. However, there are no public datasets available of cutting slope images for segmentation task. The image dataset used in this paper is captured by a UAV (DJI spirit 4 Pro v2.0, DJI, Shenzhen, China). The 100 original slope images were obtained by photographing several cutting slopes of different levels in a section of the Shenzhen-Cenxi highway in Guangdong Province, China. The slope images had a resolution of 5472 × 3078 and an average shooting height of 118 meters; all shot under clear weather and good lighting. In addition to slope components, the images also contain other less relevant scenes (i.e., roads, vegetation, sky, etc.). A total of 5000 cutting slope scenes with a resolution of 960 × 720 were obtained from the 100 high-resolution images using a sliding window (960 × 720) and step size of 500 (Fig. 7). This process not only increases the proportion that the slope components occupy to optimize labeling, but also increases the number of samples for training or testing.

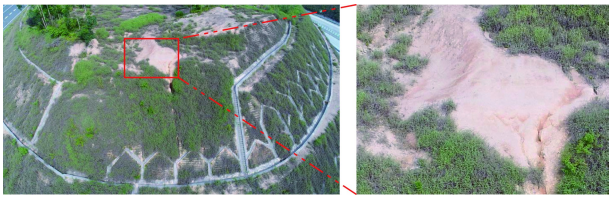


Fig. 7 Raw image (5472 × 3078) and processed image (960 × 720).

Since many of the 5000 cutting slope images show repetitiveness or similarities, 971 different, high quality images were selected for training and testing. Among them, 903 were randomly selected as the training samples, while the remaining 68 were used as the testing samples. To test the performance of the network with rich or complex scenery, the pixels in the training sample images were classified into seven categories: lattice beam, vegetation, vegetation disappearance, sky, retaining wall, road, and signs. It should be noted that the edges of the objects of interest were manually labeled, thus the contours in the ground truth and raw image do not match 100%. The Imagelabeler toolbox of MATLAB (MathWorks Inc., Natick, MA, USA) was used to label the pixels.

2.3.2 Hyperparameter

In this paper, the training parameters are kept constant for U-Net, FCN 32s, and DeepLab v3+ with Resnet18. Two optimizers, Sgdm and Adam, are used for the three networks. With the Sgdm optimizer, the momentum parameter is 0.9, minimum batch selection is 10, maximum epoch is 30, and the whole training process has 2700 iterations to update the network weights. Since small and decreasing learning rates are recommended [38], the initial learning rate is 0.001, the change factor is 0.3, and the variation interval is 10 epochs (Fig. 8). In addition, an L2 regularization term is added to the cross-entropy loss function to reduce the overfitting effect [39].

Since the proportion of certain categories is relatively small in the training set (i.e., road signs, roads, etc.), median frequency balancing, a technique to weigh the sum of cross-entropy loss for each class to compensate for data distribution imbalance [40], is used. And the weight coefficients of each category in training are shown in Table 2. In addition, the dataset is shuffled randomly after every iteration during the training process. The software used is MATLAB R2020a and the testing process was performed on a computer with Intel (R) Xeon (R) CPU e5-2620-v4 CPU, RAM 128G.

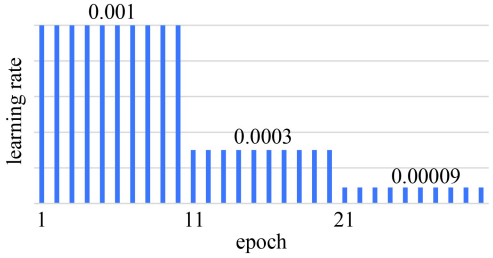


Fig. 8 Learning rate change chart.

Table 2 The weight coefficients of the cross-entropy loss function in training

LN [a]	$w(l)$
LB [b]	0.6390
VT [c]	0.1292
VD [d]	1.3671
sky	2.8740
RW [e]	0.5869
road	1
signs	20.5913

Notes: a) LN: label name; b) LB: Lattice beam; c) VT: Vegetation; d) VD: Vegetation Disappearance; e) RW: Retaining Wall.

2.4 Additional evaluations

To evaluate the reliability and robustness of the model, the optimal network model will be verified in three additional aspects.

1) K-fold cross validation experiment: The data is divided into K parts with only one taken as the testing set each time. The rest is used as the training set to train the network model. This process is repeated K times until each part of the dataset is used as a test set. In this paper, K = 10 (10 has been widely used in relevant studies [41]).

2) AdeDelta optimizer with piecewise constant decay and cosine decay is used for training.

3) Three new sets of different cut slope images will be tested.

3 Testing images and discussion

In this section, the 68 images of cutting slope scenes mentioned above are used as the test set and the evaluation indices mentioned in Section 2.2 are used to evaluate the performance of the trained semantic segmentation network. The objectives are:

1) to compare the performance of three semantic segmentation networks with cutting slope scenes;

2) to compare the performance of two optimizers, Sgdm and Adam;

3) to implement additional evaluations on the optimal network model.

3.1 Testing image results

A total of 68 images sized 960 × 720 × 3 are used to test the performance of the 3 trained semantic segmentation networks. These images have not been used in training. The test results are shown in Tables 3−6 and the partial prediction results are presented in Fig. 10. The Convergence graphs (loss/accuracy vs. number of epoch) of the training and validation datasets for the CNN model, as well as the confusion matrix for the classification metric with presented Pixel level classification network models are shown in Figs. 20 and 21 of Appendix.

Table 3 Precision of testing images of cutting slopes (%)

model	precision (*PR*)							
	LB	VT	VD	sky	RW	road	signs	MPR
U-Net (Sgdm)	NaN	75.57	NaN	NaN	NaN	NaN	NaN	10.80
U-Net (Adam)	47.51	97.25	22.84	34.73	22.05	13.00	21.29	36.95
FCN (32s Sgdm)	74.72	98.86	35.77	**92.52**	**91.17**	**91.67**	45.82	75.79
FCN (32s Adam)	48.59	97.49	13.29	NaN	32.57	NaN	NaN	27.42
DeepLab v3+ (Resnet18 Sgdm)	**81.42**	**98.99**	**38.30**	88.27	82.55	85.06	**59.52**	**76.30**
DeepLab v3+ (Resnet18 Adam)	76.13	98.66	27.09	34.32	63.46	0	NaN	42.81

Note: Bold font is the best case.

Table 4 Per-class pixel accuracy of testing images of cutting slopes (%)

model	per-class pixel accuracy (*PA*)								
	LB	VT	VD	sky	RW	road	signs	*MPA*	*GPA*
U-Net (Sgdm)	0	100	0	0	0	0	0	14.29	75.57
U-Net (Adam)	70.95	88.71	48.30	46.54	4.64	0.23	24.74	40.59	79.57
FCN (32s Sgdm)	93.81	86.46	**91.68**	93.17	96.51	92.61	77.66	90.27	88.23
FCN (32s Adam)	**94.08**	85.98	19.65	0	3.05	0	0	28.96	79.29
DeepLab v3+ (Resnet18 Sgdm)	92.89	**89.54**	82.70	**99.42**	**98.35**	**97.18**	**79.14**	**91.32**	**90.32**
DeepLab v3+ (Resnet18 Adam)	86.57	85.91	84.08	86.29	84.32	0	0	61.02	84.00

Note: Bold font is the best case.

Table 5 Intersection of union of testing images of cutting slopes (%)

model	intersection of union (*IoU*)								
	LB	VT	VD	sky	RW	road	signs	*MIoU*	*WIoU*
U-Net (Sgdm)	0	75.57	0	0	0	0	0	10.80	57.11
U-Net (Adam)	39.78	86.54	18.35	24.82	3.99	0.23	12.92	26.66	72.15
FCN (32s Sgdm)	71.21	85.61	34.64	86.65	**88.26**	**85.42**	40.48	70.32	81.79
FCN (32s Adam)	47.15	84.11	8.61	0	2.87	0	0	20.39	70.74
DeepLab v3+ (Resnet18 Sgdm)	**76.64**	**88.73**	**35.46**	**87.81**	81.43	83.01	**51.45**	**72.08**	**84.71**
DeepLab v3+ (Resnet18 Adam)	68.09	84.92	25.77	32.55	56.76	0	0	38.30	77.09

Note: Bold font is the best case.

Table 6 Mean boundary F1 score of class of testing images of cutting slopes (%)

model	mean boundary F1 score of class (*MBFSoC*)							
	LB	VT	VD	sky	RW	road	signs	*MBFSoDS*
U-Net (Sgdm)	NaN	45.11	NaN	NaN	NaN	NaN	NaN	6.44
U-Net (Adam)	46.10	61.48	**27.12**	12.27	11.38	9.38	43.59	30.19
FCN (32s Sgdm)	55.33	61.48	15.93	**54.55**	**64.02**	**52.61**	9.36	**44.75**
FCN (32s Adam)	29.48	49.92	8.34	NaN	3.68	NaN	NaN	13.06
DeepLab v3+ (Resnet18 Sgdm)	**65.03**	**70.81**	24.68	29.61	35.79	39.29	**32.12**	42.48
DeepLab v3+ (Resnet18 Adam)	59.47	65.89	20.98	20.04	29.86	0	NaN	28.03

Note: Bold font is the best case.

Results show that five of the six situations performed well (including one U-Net (Adam), FCNs, and DeepLab v3+). Among them, DeepLab v3+ with Resnet18 Sgdm achieved the highest *GPA* (90.32%), while the FCN 32s Adam achieved the lowest *GPA* (79.29%). The U-Net Sgdm demonstrated poor performance, though scored high values in certain evaluation indices, such as *GPA* and *WIoU*. However, U-Net Sgdm mistakenly classified

all pixels in the cutting slope images as vegetation, as shown in Fig. 10(e). While the evaluation index value of U-Net Sgdm is high, the fact that it predicted all pixels as "vegetation", which often occupy a majority of the pixels of cutting slope images (Fig. 9 and Table 7), resulted in a much larger *PA* value. This, in turn, resulted in a much larger *GPA* index and plausible effect in the evaluation index.

For the program execution time (Table 8), the minimum time used by the DeepLab v3+ with Resnet18 and Sgdm model is 1.56 s/image.

Figure 10 shows the different predictions of DeepLab v3+ with Resnet18 and Sgdm, FCN 32s Sgdm, and U-Net Adam. Though the evaluation index of the image prediction results of FCN 32s Sgdm was plausible, the actual prediction results do not match the actual situation shown in the cutting slope images. Specifically, curved contour features were recognized as straight lines, certain details were lost, and even the contour of ground truth was inconsistent, let alone the contour of the original image. The *MBFSoDS* index of the DeepLab v3+ combined with Resnet18 and Sgdm model is 2.27% less than that of the FCN 32s Sgdm model since FCN is more accurate when identifying objects with straight contour. Though DeepLab v3+ fits very well with the contour of the ground truth but not with the original image, it is important to keep in mind that the two often contain small, unavoidable discrepancies from human error.

The U-Net Adam model fits well with the original image. FPR in Table 9 indicates that U-Net Adam often classified pixels into the wrong class (i.e., in the fourth row of Fig. 10, the U-Net Adam model identified the entire retaining wall as a lattice beam). In addition, the performance of U-Net Adam in the evaluation index is less than ideal since it often classifies pixels of the same brightness into the same category.

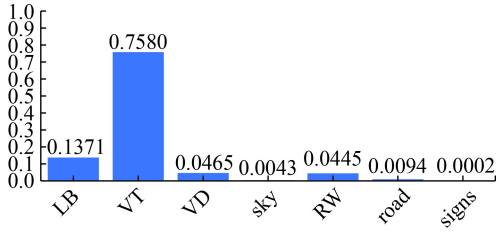


Fig. 9 Pixel distribution frequency of each category in 971 image sets.

Compared with the ground truth, the overall evaluation index of DeepLab v3+ with Resnet18 and Sgdm demonstrated the best prediction results on cutting slope images.

To verify the reliability of the DeepLab v3+ with Resnet18 and Sgdm model, 3 additional evaluations will be carried out.

First, K-fold cross validation experiment was implemented to assess the robustness of the optimal model (DeepLab v3+ combined with Resnet18 and Sgdm). Table 10 shows that after 10-fold cross validation, the average *GPA* and *MIoU* values differ from the origin by only 0.21% and 2.66%, respectively (903 images of training set and 68 images of testing set). The results show that the network model was not over-fitting and had ideal robustness.

To find suitable training hyperparameters (i.e., optimizer and learning rate decay strategy), the AdeDelta optimizer of cosine learning rate decay and piecewise constant decay are applied to the optimal network model (DeepLab v3+ with Resnet18). Results (Table 11) show that DeepLab v3+ (Resnet18) works best with the Sgdm optimizer and piecewise constant decay. It is worth mentioning that when DeepLab v3+ is combined with AdeDelta, certain small objects in the images (i.e., roads and road signs) are incorrectly classified into lattice beams, sky, and retaining walls, resulting in poor index scores.

More images from three different cutting slopes are used for further testing to determine the actual value of the optimal model (DeepLab v3+ combined with Resnet18 and Sgdm). Recognition results (Fig. 11) show that, in cases 1 and 3, the pixels of the lattice beam close to vegetation were wrongly recognized due to the color similarities. In addition, the contours of other classes are more accurate in complex and staggered scenes,

Table 7 Number of pixels by category

label	pc [a)]	ipc [b)]	real-world objects
LB	92014213	599270400	concrete lattice beam, concrete ladder, concrete drainage channel
VT	508610045	671155200	grass, trees
VD	31215511	448588800	soil exposed after vegetation disappearance
Sky	2855823	78796800	sky
RW	29833466	184550400	concrete and marble retaining walls
Road	6273698	62208000	roads, highway guardrails, road drains, traffics
Signs	126202	20044800	road signs

Notes: a) pc: pixel count, the total number of pixels in this class; b) ipc: image pixel count, the total number of image pixels containing this category.

indicating that the recognition performance of the model is not bad.

3.2 Discussion

Different segmentation networks demonstrate different pixel classification accuracies. This subsection visualizes the feature extraction and expansion processes of input image into output results within the network.

It should be noted that the presented feature maps are adjusted to the [0,1] using the MATLAB function "mat2-gray". The degree of recognition and extraction of the map features are expressed in the rainbow legend with the redder colors representing the higher degree, and vice versa. The input image is shown in Fig. 12.

The DeepLab v3+ combined with Resnet18 and Sgdm, FCN 32s Sgdm, and U-Net Adam are selected for comparative discussion.

Table 8 Network program execution time

model	PET
U-Net (Sgdm)	8.85
U-Net (Adam)	26.26
FCN (32s Sgdm)	4.10
FCN (32s Adam)	4.77
DeepLab v3+ (Resnet18 Sgdm)	**1.56**
DeepLab v3+ (Resnet18 Adam)	3.09

Note: PET = program execution time (s/image). Bold font is the best case.

3.2.1 DeepLab v3+ combined with Resnet18 and Sgdm

Figure 4 shows the network structure of Deeplab v3+ combined with Resnet18 and Sgdm. After the first activation (L4) of the image input to the network, 64

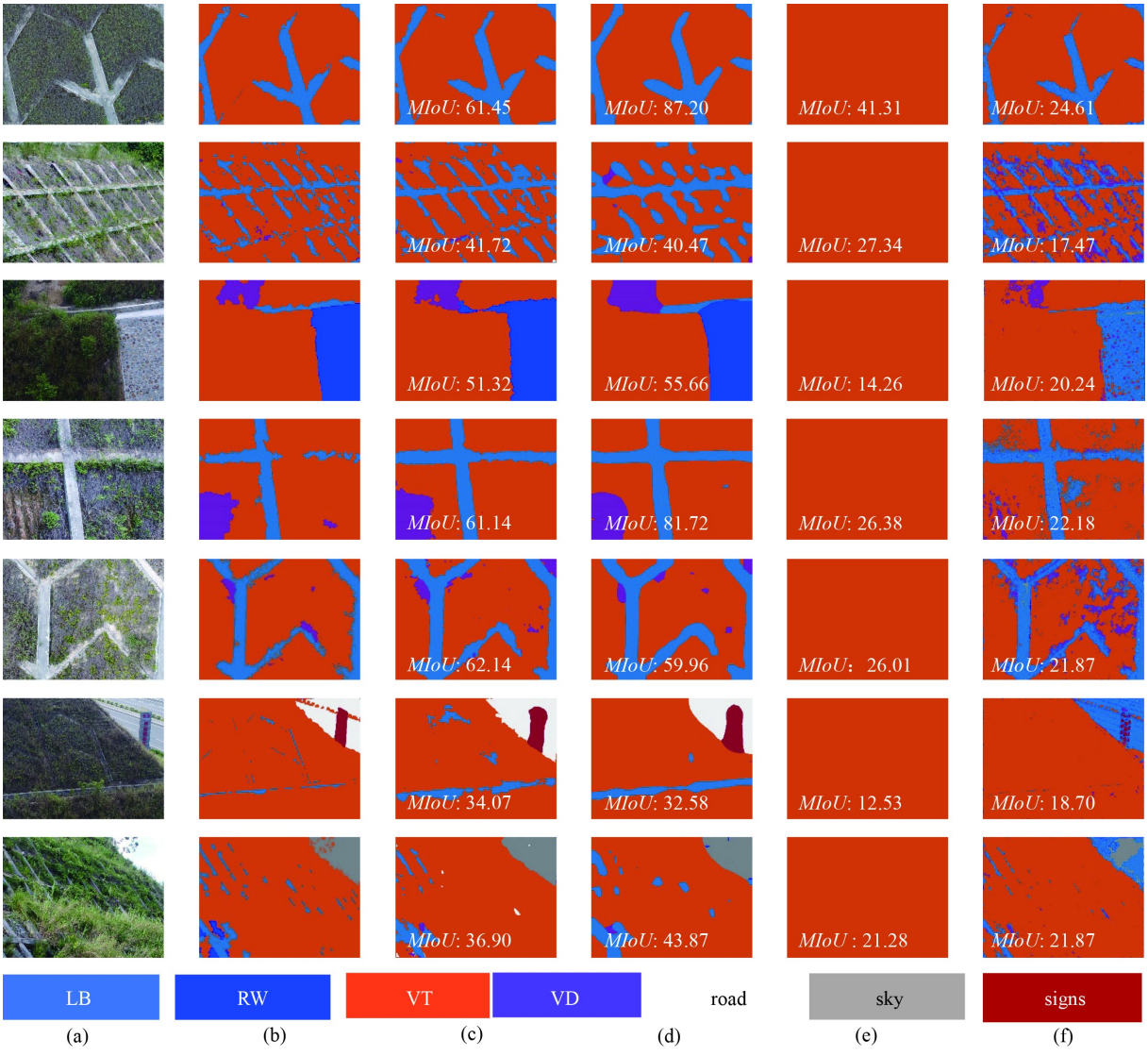


Fig. 10 Prediction results of partial semantic segmentation model. (a) Input image; (b) ground truth; (c) DeepLab v3+ combined with Resnet18 and Sgdm; (d) FCN 32s Sgdm; (e) U-Net Sgdm; (f) U-Net Adam.

Table 9 *FPR* of networks (%)

model	false positive rate (FPR)								
	LB	VT	VD	sky	RW	road	signs	MPFR	GFPR
U-Net (Adam)	13.15	7.76	5.98	0.85	0.57	**0.03**	0.08	4.06	20.43
FCN (32s Sgdm)	5.32	3.08	6.03	**0.07**	**0.33**	0.18	0.08	2.16	11.77
DeepLab v3+ (Resnet18 Sgdm)	**3.56**	**2.81**	**4.88**	0.13	0.72	0.37	**0.05**	**1.79**	**9.68**

Note: Bold font is the best project.

Table 10 The result of the K-fold cross validation (%)

Situation	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	mean	origin
GPA	91.92	93.09	94.82	95.47	91.38	94.67	86.61	87.08	88.64	91.60	91.53	91.32
MIoU	69.34	68.24	75.96	77.77	70.68	73.31	64.35	65.25	62.19	67.70	69.42	72.08

Table 11 Prediction results of DeepLab v3+ combined with different optimizer and learning rate decay strategy (%)

DeepLab v3+ (Resnet18)	index						
	MPR	*MPA*	*GPA*	*MIoU*	*WIoU*	*MBFSoDS*	PET
Sgdm and PCD	**76.30**	**91.32**	**90.32**	**72.08**	**84.71**	**42.48**	**1.56**
AdeDelta and PCD	13.72	40.44	78.79	20.40	73.96	22.91	3.67
AdeDelta and cosine decay	13.29	30.25	70.04	16.43	65.93	17.68	1.89

Note: Bold font is the best case. PCD = piecewise constant decay; PET = program execution time (s/image).

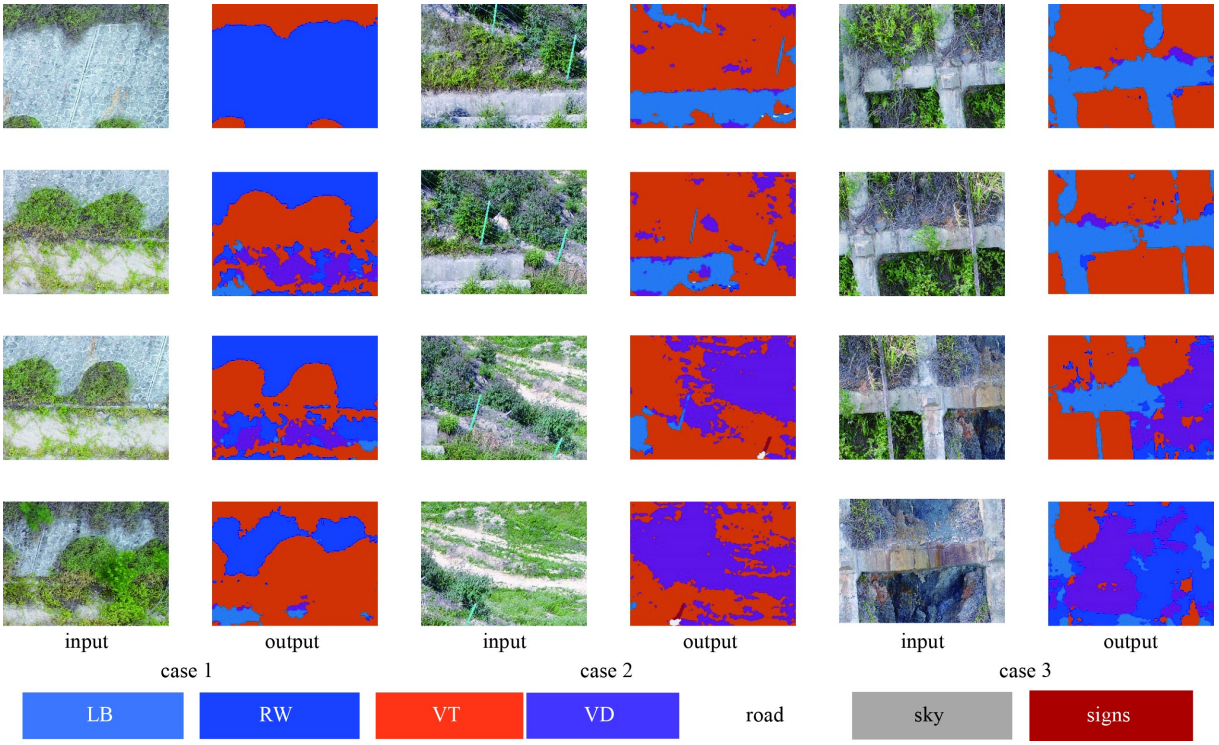


Fig. 11 DeepLab v3+ (combined with Resnet18 and Sgdm) in 3 cases.

feature maps are obtained, shown in Fig. 13, where L4-22 represents the 22nd feature map obtained after layer 4 is activated.

Due to the large number of feature maps (as the number of feature maps increase, the resolution becomes lower as it reaches the end of the feature extraction part), only certain feature maps of noticeable significance are selected for discussions. The results are shown in Fig. 14.

Take the lattice beam in the image as an example. Along the feature extraction path, as the number of

Fig. 12 Input image (raw image).

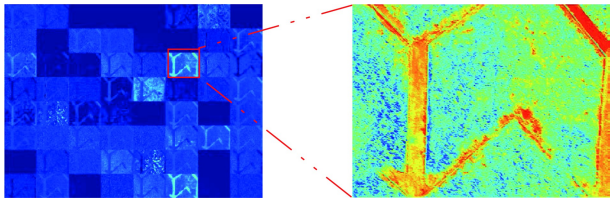


Fig. 13 Among the 64 feature maps after L4 activation, the 22nd feature map is extracted.

network layers increases, the size of the feature maps decreases, making the contour of the lattice beam more representative of the original delicacy. After multiple layers of feature extraction, the comparison between L4-22 and L67-405 (Fig. 14) shows that the outer contour of the lattice beam has become ambiguous and is replaced by an approximate shape.

In Fig. 14, L70, L73, L76, and L79 each have 256 feature maps after 4 parallel AC operations, which is used to increase the layer's receptive field without increasing the number of parameters or computation [20]. The feature maps obtained from AC operations contain representative features of the input image. Thus, the results (L83-25) of feature extraction operation not only demonstrate the features of the lattice beam, but also eliminate other irrelevant features.

In Fig. 14, L84 is a transposed convolution layer. The comparison between L84-25 and the previous layer shows an increase in image resolution and lattice beam features. The feature maps of L95 are obtained by a DC operation on the shallow feature maps (L87) and deep

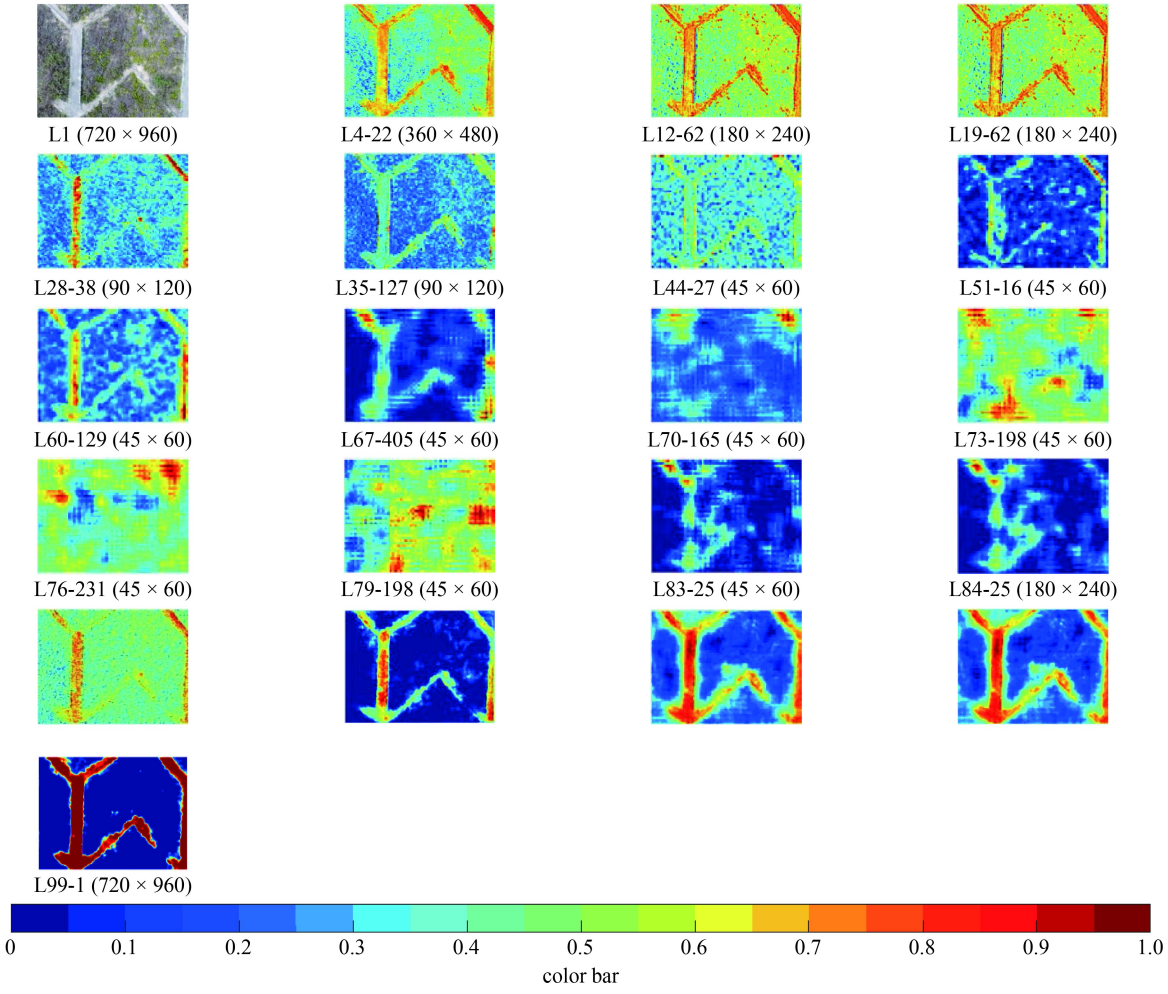


Fig. 14 Visualization results of feature map of input image by DeepLab v3+ combined with Resnet18 and Sgdm network.

feature maps (L84), followed by a feature extraction operation. The feature contour of the image target extracted from the front part of the network is more identifiable, while the one extracted from the back part is more representative of the original image. Since the general semantic segmentation network has deep layers, many detailed contours may have been lost after deep feature extraction, which even the transposed convolution operation may not be able to restore. Therefore, the deep layer feature maps require deep concatenation with the delicate feature maps from shallow layers to produce combined feature maps that contain fine contour features. The extraction results of these feature maps show clear lattice beam features (L95-33). In other words, feature extraction after "deep concatenation" not only extracts the lattice beam features from the deep feature maps (such as L84-25) and discards irrelevant information, but also obtains lattice beam details from the shallow feature maps (such as L87-4). The results are shown in L95-33 (Fig. 14).

The feature maps of L96 are obtained from the feature maps of L95 through seven convolution kernels sized 1 × 1, which functions to classify pixels of feature maps. Then, the transposed convolution layer expands the feature maps of the previous layer (L96) into the same resolution as the input image. Finally, the Softmax layer converts the values in the L97 feature maps into probability form and the final layer attaches the category label to the corresponding position of the feature map according to the standards mentioned above. The above process describes pixel level classification of the DeepLab v3+ with Resnet18, which has 18 layers of deep feature extraction.

Since additional feature extraction layers may lead to better classification, a DeepLab v3+ with Resnet50 model with 50 layers deep feature extraction was tested (Resnet18 is replaced by Resnet50 and the remaining tail structure of DeepLab v3+ is kept). In terms of the evaluation indices *PR*, *PA*, *IoU*, and *MBFSoC*, DeepLab v3+ with Resnet50 performed slightly better than DeepLab v3+ with Resnet18 (Table 12). However, the computational cost would increase significantly for certain network series [16]. Since the network structure of DeepLab v3+ with Resnet50 is more complex than that of DeepLab v3+ with Resnet18, the image prediction process also takes longer. It is also worth mentioning that the training time of DeepLab v3+ with Resnet18 is 45% more than that with Resnet50 due to its fewer network weight parameters that require adjusting. Thus, DeepLab v3+ with Resnet18 is more practical for cutting slope image recognition.

3.2.2 FCN 32s Sgdm

Section 3.2.2 discusses feature map visualization for FCN 32s Sgdm. In comparison, the object contours of the cutting slope scenes predicted by FCN 32s Sgdm are relatively straight with the original rugged contour replaced by smooth lines. Similar to Section 3.2.1, only selected feature maps are discussed. The feature maps of each layer are shown in Fig. 15.

In the first convolution of the FCN 32s semantic segmentation model, the width and height are padded with 0 and the padding size is 198. This operation of filling pixels outward makes the contour of the object of interest more centered in the feature map, which can increase the chances of the network extracting the key components of the feature maps.

The feature maps of L10 are obtained from L5 after multiple feature extractions (Fig. 15). Interestingly, the feature extraction path of this part (L6–L10) not only extracts the position of vegetation class in the input image (L10-47), but also deliberately extracts the information of the lattice beam edge (L10-2). Figure 15 clearly indicates in L17-103 that the activated position in the feature map forms the outer edge contour of the lattice beam. It can be seen from L24-47 that the outer edge contour feature of the lattice beam is extracted. From another point of view, the outer contour corresponding to the vegetation position is also clearly divided. The maximum pool layer, L25, reduces the size of the feature maps. The comparison between L24-388 and L25-388 shows an obvious difference in the resolution between these two. Furthermore, L39 is obtained from 4096 L38 feature maps

Table 12 Prediction results of DeepLab v3+ network with two different feature extraction layers (%)

Model	Index						
	MPR	*MPA*	*GPA*	*MIoU*	*WIoU*	*MBFSoDS*	PET
DeepLab v3+ (Resnet18 Sgdm)	76.30	91.32	90.32	72.08	84.71	42.48	**1.56**
DeepLab v3+ (Resnet18 Adam)	42.81	61.02	84.00	38.30	77.09	28.03	3.09
DeepLab v3+ (Resnet50 Sgdm)	**77.61**	**91.90**	**91.63**	**73.71**	**86.16**	**47.56**	2.87
DeepLab v3+ (Resnet50 Adam)	32.88	42.92	82.60	28.18	75.36	21.87	15.12

Note: Bold font is the best case.

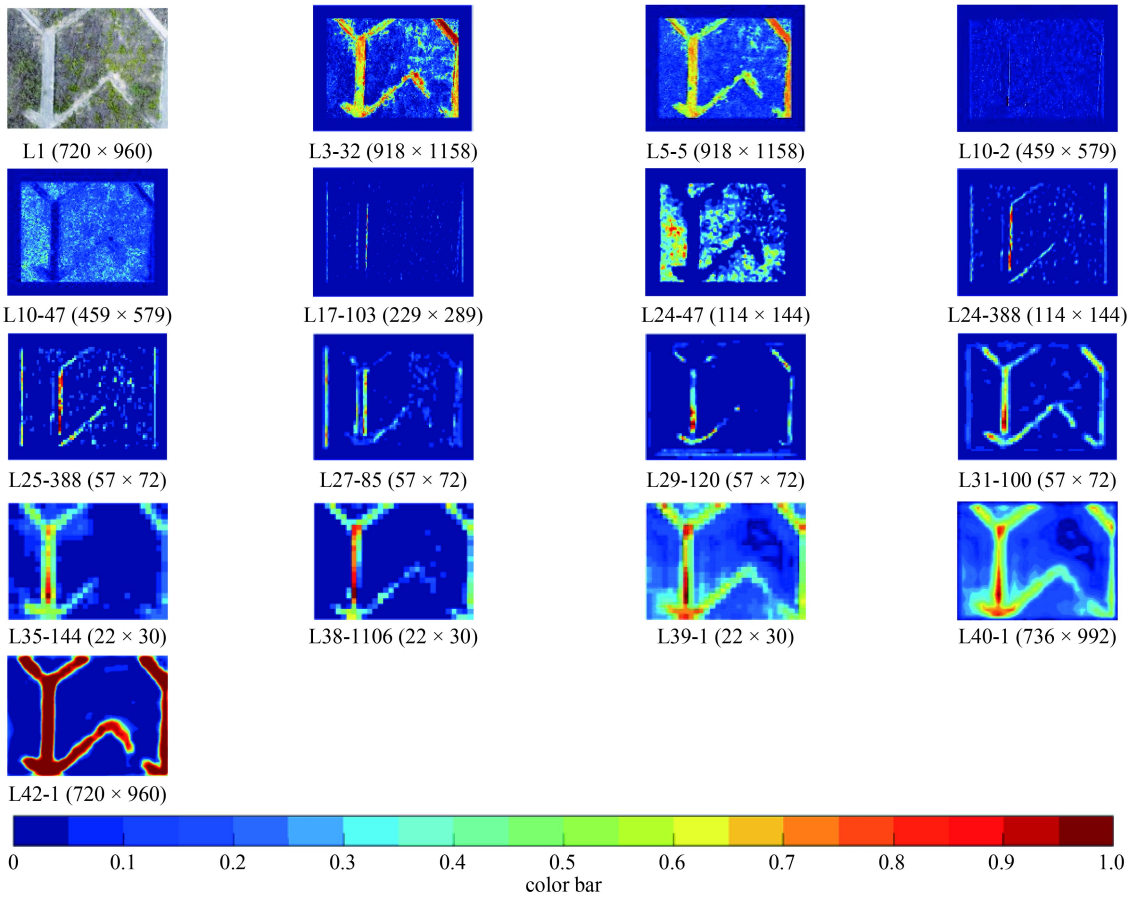


Fig. 15 Visualization results of feature map of input image by the FCN 32s Sgdm network.

through 7 convolution kernels of 1 × 1. The contour of the lattice beam has been completely and smoothly extracted. The high-resolution image (L40-1) is obtained after the feature map passes through the transposed convolution layer and its values converted into probability form (L42-1) through the Softmax layer. Finally, the last layer labels the pixels with their corresponding probability values.

Compared to DeepLab v3+ with Resnet18 and Sgdm, FCN 32s Sgdm only uses one transposed convolution layer, thus the results are much smoother since it is difficult to directly restore the contour information of the lattice beam with fine discrimination from low resolution. However, this operation is conducted by DeepLab v3+ with Resnet18 twice so that the network not only expands image resolution, but also extracts additional details. Since FCN 32s does not use “deep concatenation”, the program loses accurate contour information of the object of interest, making DeepLab v3+ with Resnet18 better at feature restoration.

3.2.3 U-Net Adam

Section 3.2.3 discusses the visualization of feature maps of U-Net Adam by comparing it with the previous two models. L5-56 of Fig. 16 is one of the feature maps of the input image L1 after 2 convolution layers, 1 activation layer, and profile feature extraction of the lattice beam.

In Fig. 19, the profile features of the lattice beam can still be extracted by the network even after the resolution of L10-33 is reduced. In L15, 256 feature maps are generated, but none extracted the features of the lattice beam (Fig. 17).

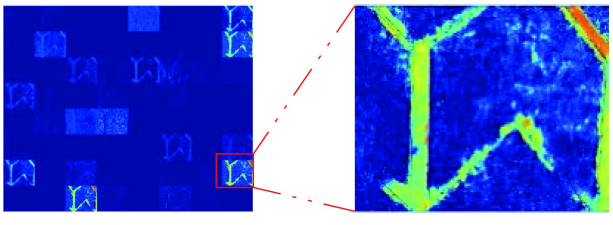


Fig. 16 The 64 feature maps of L5 in U-Net Adam and L5-56.

Fig. 17 The 64 feature maps of L15 in U-Net Adam.

In Fig. 19, the features of other objects in L15–L43 were extracted, rather than that of the lattice beam. In the network structure, the deep layers (L29 and L36) are concatenated with the shallow layers (L20 and L15), producing L30 and L37, respectively. After the feature extraction process, L34 and L50 were unable to extract features of the lattice beam, unlike DeepLab v3+ combined with Resnet18 and Sgdm. However, the lattice beam features (L55-6) sudden appeared in L55 after feature extraction due to "deep concatenation" between L50 (deep layer) and L5 (shallow layer). As mentioned previously, none of the feature maps in L15 had obvious lattice beam features, making it difficult for the network to extract its contour. This is because the values of the lattice beam positions in the L15 feature map were mapped to 0 by the Relu layer and applying convolution operation (feature extraction) to L15 cannot make the values of these positions (the pixel of the lattice beam) greater than 0, thus the outcome is the same as L30.

It should be noted that the feature maps of L48 were obtained after L43 (without the lattice beam contour) and L10 (with the lattice beam contour) are concatenated and underwent feature extraction. However, the position of the lattice beam was still not activated in the 128 feature maps of L46 (Fig. 18) because the convolution layer is more inclined to extract the features of non-lattice beams. In other words, the feature vectors of the lattice beam contour in the feature maps all have negative values.

In Fig. 19, L29, L36, L43, and L50 are the feature maps obtained after the transposed convolution feature expansion. Among them, L56 was obtained from 64 feature maps of L55 through 7 convolution kernels of 1 × 1. L57-1 is the result of the Softmax layer.

In terms of the entire U-Net model, the L2–L10 layers function to extract features of the lattice beams, while the L11–L50 layers function to extract features of other elements of interest. The features of lattice beams in L51–L55 were extracted from the fine shallow feature maps and rough deep feature maps. Finally, each pixel is labeled with their corresponding probability value.

Prediction results show that the contour extracted by U-Net is closest to the input image, though the network is more vulnerable to pixel interference with the same brightness, such as when vegetation disappearance is incorrectly classified as the lattice beam. The darker part is also prone to error and easily misclassified as

Fig. 18 128 feature maps of L46 in U-Net Adam.

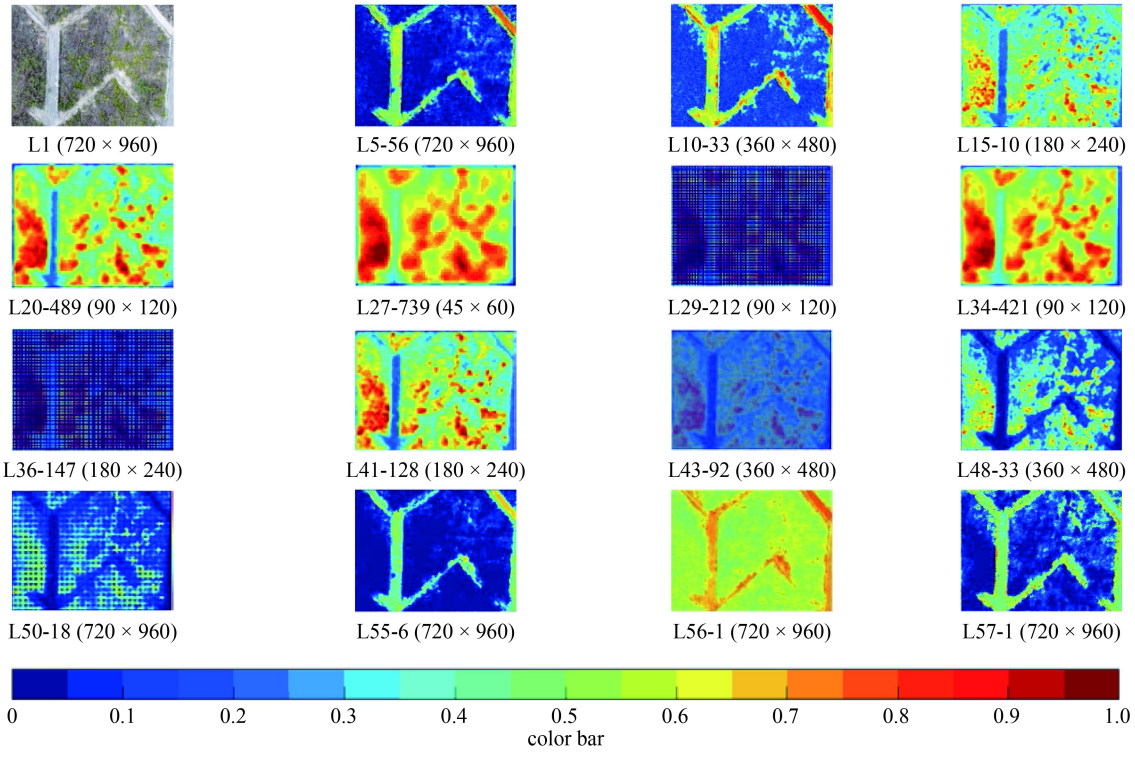


Fig. 19 Visualization results of feature maps of input image by the U-Net Adam network.

vegetation. In Table 9, the FPR of the U-Net is the highest, thus the score on the evaluation index is not as good as the other two.

4 Conclusions

Finding an appropriate semantic segmentation network to predict the contour of elements of interest for slope images (i.e., lattice beams, vegetation disappearance, retaining walls, etc.) establishes the foundation of mapping slope surfaces and evaluating slope safety and stability. This paper uses three semantic segmentation networks combined with two optimizers, Sgdm and Adam. Finally, the prediction results are discussed from the perspectives of feature extraction, expansion, and feature map visualization.

With the above research, the following conclusions can be drawn.

1) From the evaluation indices we can see that the best semantic segmentation network is DeepLab v3+ with the highest MPR value with the Resnet18 feature extractor and Sgdm optimizer. At the same time, DeepLab v3+ also achieved the highest scores in *GPA* and *WIoU*. The program execution time is shortest when the feature extractor is Resnet18 and Sgdm is the optimizer. DeepLab v3+ combined with Resnet18 and Sgdm performed the best in terms of cost performance.

2) For the FCN 32s model, the Sgdm optimizer performed better than the Adam optimizer.

3) The U-Net model performed poorly except when combined with the Adam optimizer. The MPR, *GPA*, *WIoU*, and *MBFSoDS* of the U-Net Adam model are 10.75%−39.35% lower than that of DeepLab v3+ combined with Resnet18 and Sgdm. *PR* had the biggest difference of 39.35%, while *GPA* had the smallest difference of 10.75%.

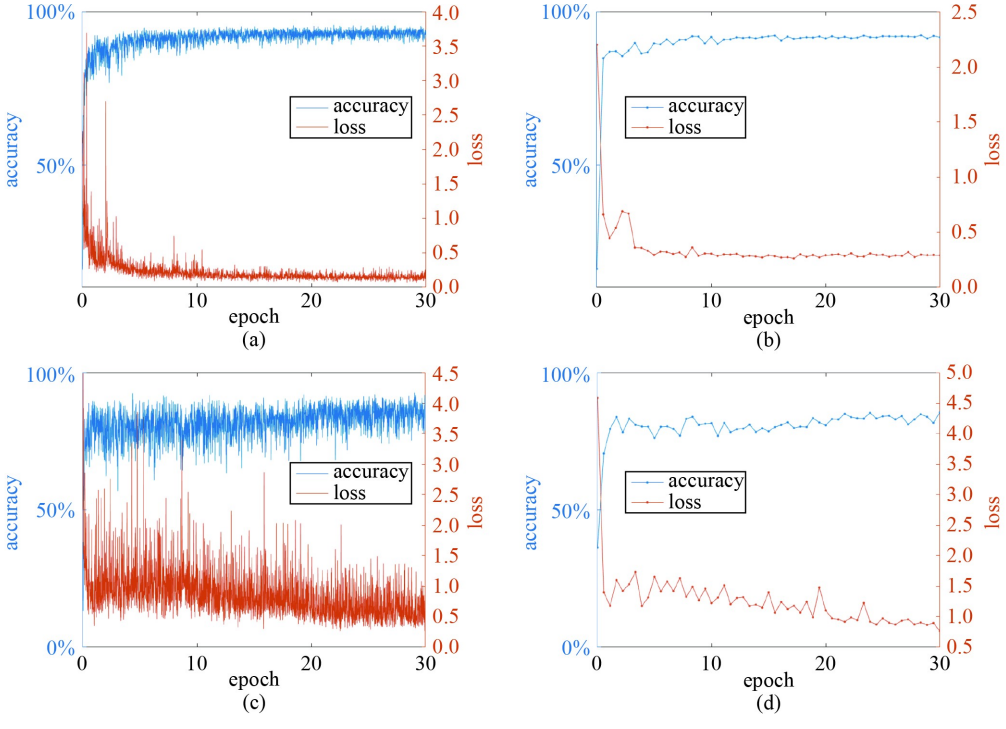
4) In feature maps visualization, the DeepLab v3+ model combined with Resnet18 and Sgdm produced results of the input image closest to the ground truth. The FCN 32s model often ignored details of the objects of interest and does not reflect the real contour. The prediction results of U-Net is closest to the input image, but it is still prone to errors (*FP* in Table 9 is high), resulting in poor evaluation indices.

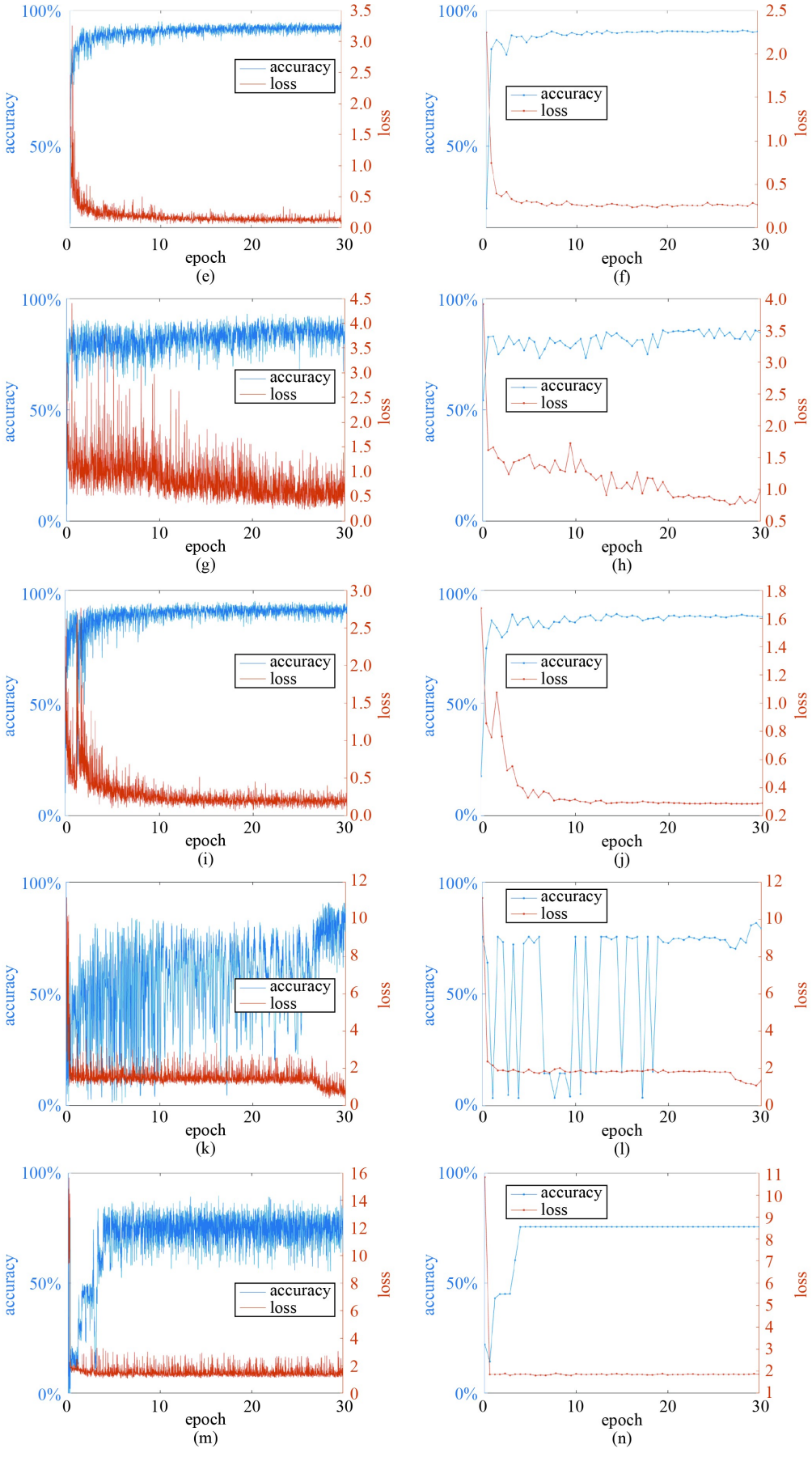
An appropriate, practical network model can lay the foundation for slope image recognition and change quantification. The statistical information of the number of pixels of the elements concerned by semantic segmentation can be used as a reference index for the slope safety.

Appendix

See Figs. 20 and 21.

Acknowledgements The authors would like to express their sincere gratitude to Yang HE, Wei DENG, Ronghao ZHANG, from the Guangdong University of Technology, for labeling the image data.




accuracy
loss
epoch
(e)
(f)
(g)
(h)
(i)
(j)
(k)
(l)
(m)
(n)

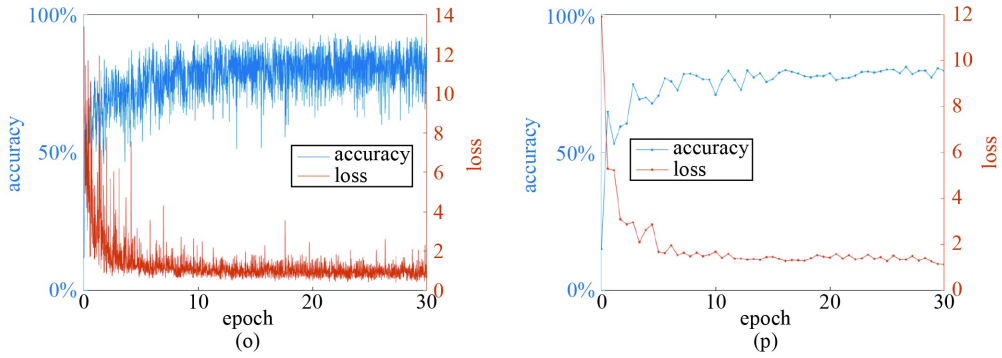
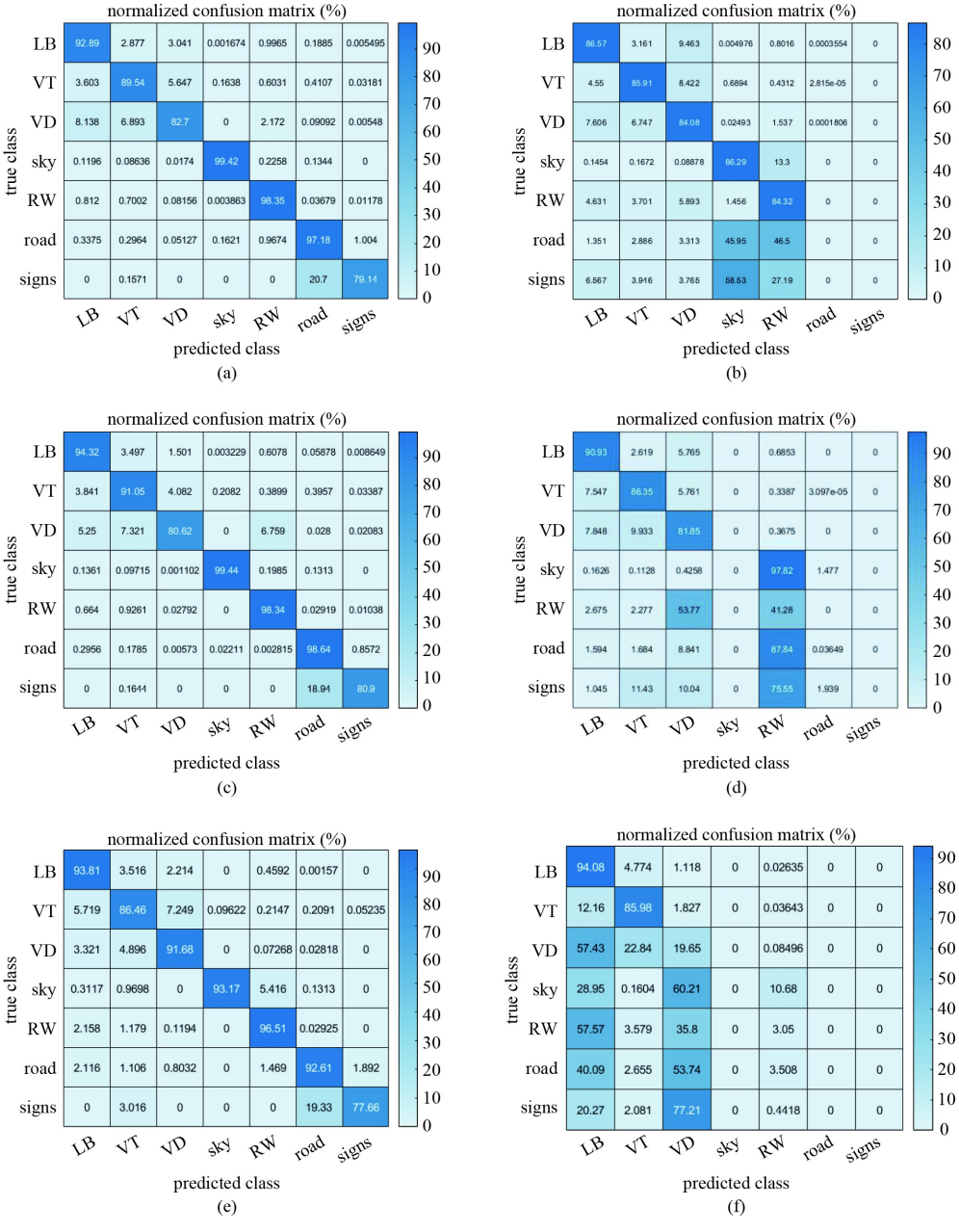


Fig. 20 Convergence graphs (loss/accuracy vs number of epoch) on training and validation dataset for the CNN model. TD = training dataset. VD = validation dataset.



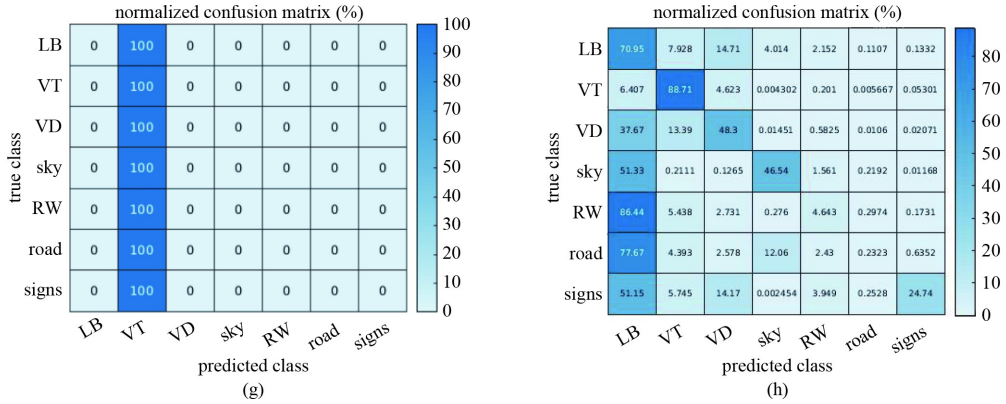


Fig. 21 Confusion matrix for the classification metric with presented Pixel level classification network models. (a) DeepLab v3+ (Resnet18 Sgdm); (b) DeepLab v3+ (Resnet18 Adam); (c) DeepLab v3+ (Resnet50 Sdgm); (d) DeepLab v3+ (Resnet50 Adam); (e) FCN 32s (Sgdm); (f) FCN 32s (Adam); (g) U-Net (Sdgm); (h) U-Net (Adam).

References

1. Chen Z. Soil Slope Stability Analysis—Principle, Methods and Programs. Beijing: China Water & Power Press, 2003 (in Chinese)
2. Wu C I, Kung H Y, Chen C H, Kuo L C. An intelligent slope disaster prediction and monitoring system based on WSN and ANP. Expert Systems with Applications, 2014, 41(10): 4554−4562
3. Shu J, Zhang J, Wu J. Research on highway slope disaster identification based on deep convolution neural network. Highway Traffic Technology, 2017, 13(10): 70–74 (in Chinese)
4. Wu J. Feature learning of highway image and detection of slope failure. Thesis for the Master's Degree. Beijing: Beijing University of Posts and Telecommunications, 2018 (in Chinese)
5. Xu J, Gui C, Han Q. Recognition of rust grade and rust ratio of steel structures based on ensembled convolutional neural network. Computer-Aided Civil and Infrastructure Engineering, 2020, 35(10): 1160−1174
6. Guo H, Zhuang X, Rabczuk T. A deep collocation method for the bending analysis of Kirchhoff plate. Computers, Materials & Continua, 2019, 59(2): 433−456
7. Anitescu C, Atroshchenko E, Alajlan N, Rabczuk T. Artificial neural network methods for the solution of second order boundary value problems. Computers, Materials & Continua, 2019, 59(1): 345−359
8. Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh V M, Guo H, Hamdia K, Zhuang X, Rabczuk T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. Computer Methods in Applied Mechanics and Engineering, 2020, 362: 112790
9. Zhou H, Chen Y, Tian R. Distance prediction of slope-foot landslide in southwest of China based on GA-BP neural network. In: 2019 the 6th Annual International Conference on Material Engineering and Application. Guangzhou: IOP Publishing, 2020
10. Xing Y, Wang J, Li X, Liu R, Gao J. Slope stability prediction model based on GA-SVM. In: 2010 International Conference on Educational and Information Technology. Chongqing: IEEE, 2010
11. Lin H M, Chang S K, Wu J H, Juang C H. Neural network-based model for assessing failure potential of highway slopes in the Alishan, Taiwan Area (China): Pre- and post-earthquake investigation. Engineering Geology, 2009, 104(3-4): 280−289
12. Xia Y, Chen B, Weng S, Ni Y Q, Xu Y L. Temperature effect on vibration properties of civil structures: A literature review and case studies. Journal of Civil Structural Health Monitoring, 2012, 2(1): 29−46
13. Yao X. Evolutionary artificial neural networks. International Journal of Neural Systems, 1993, 4(3): 203−222
14. Lin Y, Nie Z, Ma H. Structural damage detection with automatic feature-extraction through deep learning. Computer-Aided Civil and Infrastructure Engineering, 2017, 32(12): 1025−1046
15. Zhong K, Teng S, Liu G, Chen G, Cui F. Structural damage features extracted by convolutional neural networks from mode shapes. Applied Sciences (Basel, Switzerland), 2020, 10(12): 4247−4262
16. Teng S, Liu Z, Chen G, Cheng L. Concrete crack detection based on well-known feature extractor model and the YOLO_v2 network. Applied Sciences (Basel, Switzerland), 2021, 11(2): 813−825
17. Ghorbanzadeh O, Meena S R, Blaschke T, Aryal J. UAV-based slope failure detection using deep-learning convolutional neural networks. Remote Sensing, 2019, 11(17): 2046−2069
18. Badrinarayanan V, Kendall A, Cipolla R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(12): 2481−2495
19. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). Munich: Springer, 2015: 234−241
20. Chen L C, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: European Conference on Computer Vision (ECCV). Munich: Springer, 2018: 833−851
21. Shelhamer E, Long J, Darrell T. Fully Convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(4): 640−651
22. Narazaki Y, Hoskere V, Hoang T A, Fujino Y, Sakurai A, Spencer B F Jr. Vision-based automated bridge component recognition with

high‑level scene consistency. Computer-Aided Civil and Infrastructure Engineering, 2020, 35(5): 465−482
23. Liu J, Yang X, Lau S, Wang X, Luo S, Lee V C S, Ding L. Automated pavement crack detection and segmentation based on two-step convolutional neural network. Computer-Aided Civil and Infrastructure Engineering, 2020, 35(11): 1291−1305
24. Dung C V, Anh L D. Autonomous concrete crack detection using deep fully convolutional neural network. Automation in Construction, 2019, 99: 52−58
25. Teng S, Chen G, Gong P, Liu G, Cui F. Structural damage detection using convolutional neural networks combining strain energy and dynamic response. Meccanica, 2020, 55(4): 945−959
26. Rojahn C, Bonneville D R, Quadri N D, Phipps M T, Ranous R A, Russell J E, Staehlin W E, Turner Z. Postearthquake Safety Evaluation of Buildings. Redwood City, CA: Applied Technology Council, 2005
27. Noh H, Hong S, Han B. Learning deconvolution network for semantic segmentation. In: 2015 IEEE International Conference on Computer Vision (ICCV). Las Condes: IEEE, 2015: 1520−1528
28. Dong C, Loy C C, Tang X. Accelerating the super-resolution convolutional neural network. In: European Conference on Computer Vision (ECCV). Amsterdam: Springer, 2016: 391−407
29. Nguyen-Thanh V M, Anitescu C, Alajlan N, Rabczuk T, Zhuang X. Parametric deep energy approach for elasticity accounting for strain gradient effects. Computer Methods in Applied Mechanics and Engineering, 2021, 386: 114096
30. Zhuang X, Guo H, Alajlan N, Zhu H, Rabczuk T. Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning. European Journal of Mechanics. A, Solids, 2021, 87: 104225
31. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV: IEEE, 2016: 770−778
32. Chen L C, Papandreou G, Kokkinos I, Murphy K, Yuille A L. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(4): 834−848
33. Cha Y J, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil and Infrastructure Engineering, 2017, 32(5): 361−378
34. He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: 2015 IEEE International Conference on Computer Vision (ICCV). Las Condes: IEEE, 2015: 1026−1034
35. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015, arXiv:1502.03167
36. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 2014, 15(1): 1929−1958
37. Csurka G, Larlus D, Perronnin F. What is a good evaluation measure for semantic segmentation? In: Proceedings of the British Machine Vision Conference. Bristol: BMVA, 2013
38. Randall Wilson D, Martinez T R. The need for small learning rates on large problems. In: International Joint Conference on Neural Networks. Washington, D.C.: IEEE, 2001: 115−119
39. Krogh A, Hertz J A. A Simple Weight Decay Can Improve Generalization. In: Proceedings of the 4th International Conference on Neural Information Processing Systems (NIPS). Denver: MIT Press, 1991
40. David Eigen R F. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: IEEE International Conference on Computer Vision (ICCV). Las Condes: IEEE, 2015,
41. Zhang Y, Yang Y. Cross-validation for selecting a model selection procedure. Journal of Econometrics, 2015, 187(1): 95−112