IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Integrative Few-Shot Classification and Segmentation for Landslide Detection

DAT TRAN-ANH[1§], BAO BUI-QUOC[2§], ANH VU-DUC[1], TRUNG-ANH DO[1], HUNG NGUYEN-VIET[1], HOAI-NAM VU[1] AND CONG TRAN[1]

[1]Posts and Telecommunications Institute of Technology, Hanoi, Vietnam
[2]Hanoi University of Science and Technology, Hanoi, Vietnam

Corresponding author: Cong Tran (e-mail: congtt@ptit.edu.vn).

This work was supported by the Qualcomm and PTIT Research Collaboration grant funded by Qualcomm (SOW Number POS-459341).

ABSTRACT There has been an ongoing demand for monitoring landslides due to the heavy economic losses and casualties caused by such natural disasters. In this paper, we introduce a swift landslide detection system that can detect and segment landslides occurring on roads. To tackle the challenges of data collection, we propose an automatic annotation procedure to create a new landslide dataset consisting of 2963 images, termed the LandslidePTIT dataset. Additionally, we construct a novel deep-learning architecture that can perform both classification and segmentation tasks well from a few annotated images of landslides. Specifically, the model consists of four main modules that are delicately designed to solve the few-shot segmentation problem using landslide images, namely hypercorrection construction, attentive squeeze block, a cross-feature layer, and broadcast and squeeze layer. Experimental results exhibit the superiority of the proposed method in comparison with competitive baselines, in terms of both quantitative and qualitative manners.

INDEX TERMS Data generation, few-shot learning, few-shot segmentation, image segmentation, landslide detection.

I. INTRODUCTION

LANDSLIDES, typically a consequence of climate change [1] and urban expansion [2], are one of the most common natural disasters today and cause severe troubles to human life and infrastructures all over the world. For example, a severe landslide occurred in mid-2020 in Vietnam, causing dozens of deaths.[1] Landslides cause roads to be blocked, which causes hurdles not only in the traffic flow but also generate various traffic problems in the form of congestion [3]. Therefore, there is a need to detect and warn of landslides as quickly as possible to identify proper counter-measures so that possible unfortunate consequences can be avoided.

Previous studies deal with two related problems of landslides, termed landslide detection and landslide prediction, in which the methods built upon machine learning and deep learning are quite common [4], [5]. However, practical applications of such methods are often limited due to the fact that deep learning models often require enormous labeled data to work well while landslides may only occur several times a year, thus it may take years to collect a sufficient amount of data. Besides, as landslides tend to occur in mountainous areas, collecting landslide images is difficult due to the danger as well as the lack of equipment in such underdeveloped areas. Several prior works tackle this problem by relying on images captured by satellites [6], [7], which are however unable to respond promptly to a real-life landslide event unless we own the satellites.

In our work, we introduce a landslide monitoring system that can detect quickly whether there is any landslide occurring on roads. Fig. 1 illustrates our newly designed system consisting of four layers, namely collection, pre-processing, cloud, and application layers. Specifically, the system deploys specialized drones to collect images on roads in different areas, inspired by several recent novel methods that enables an energy-efficient routing schedule for capturing road images by drones [8], [9]. As one of our main contributions, we propose a data generation procedure, where the generated images are pre-processed and annotated automatically. The generated images are fuel to train a deep learning-based artificial intelligence (AI) model in the cloud

[§]Equal contribution
[1]Source: https://blogtuan.info/2022/05/20/serious-landslide-on-the-route-da-lat-mui-ne/

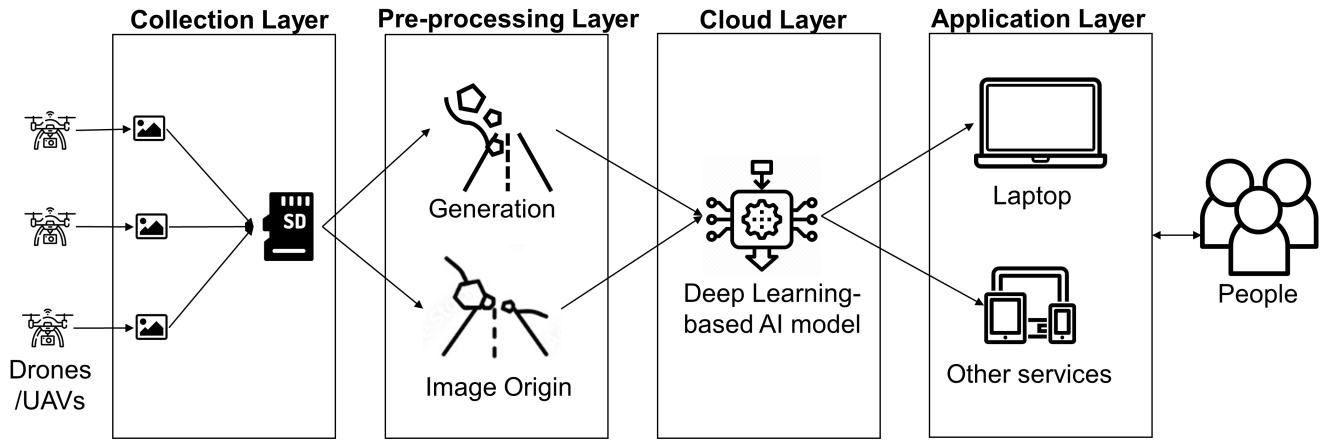


FIGURE 1: Landslide detection systems with four layers

layer that can rapidly detect landslides causing damages on roads. After heavy rain stops, the operators deploy the drones to scan the surveillance areas and transfer the images to the AI model. The locations of landslides are then detected and extracted to create a map in the application layer for immediate response of local governors. It is worth noting that even with the pre-processing layer that generates images for training, it is still challenging to train the model well due to two reasons. First, the landslide can occur in any part of the roads, making it impossible for the data generation module to exhaustively generate landslides in every possible position for training. Second, the augmented landslides from generated images may not be generalized in real practice due to different types of landslides. Thus, as another important contribution, we propose a novel detection and segmentation method based on few-shot learning. In particular, the proposed method termed Cross Feature and Attentive Squeeze Network (CF-ASNet), combines the recently state-of-the-art ASNet model with the cross-validation method. By virtue of the transferability between few-shot learning classification and few-shot learning segmentation tasks, the classification and segmentation accuracy are greatly boosted.

Our contributions are four folds and are summarized as follows:

- We introduce a new landslide detection system that aims to swiftly identify and measure the damage caused by landslides;
- We propose a novel data generation procedure, where labels of landslides are automatically assigned for training;
- We design a new model that can generalize to new types of landslides based on few-shot segmentation techniques;
- We empirically validate the effectiveness of the proposed data generation procedure as well as the newly designed model.

The organization of this paper is as follows. Section II presents relevant previous studies. In Section III, we describe the process of collecting landslide data to create our new dataset for training and testing landslides in two tasks, including detection and classification. In Section IV, we introduce a deep learning model for landslide detection based on a few-shot segmentation approach. Finally, we evaluate the proposed methods in section V.

II. RELATED WORK

Our study is related to three broad categories, namely landslide detection, landslide segmentation and few-shot learning.

A. LANDSLIDE DETECTION

Machine learning-based approaches. Machine learning techniques used in landslide detection can follow both supervised and unsupervised settings. In supervised learning, typical methods employed support vector machine (SVM) [10], [11], k-nearest neighbors (KNN) [11], Logistic Regression (LR),Random Forest (RF) [11], [12] and several other conventional classification techniques like Decision Tree [13], Naive Bayes [14], EML [15]. These studies aimed at finding the relationship between known input and unknown output to classify each image with two labels, so-called “landslide" or “non-landslide". In unsupervised methods, landslide samples are grouped based on their similarity. In [16], the authors proposed an unsupervised method by utilizing six unsupervised well-known methods, including K-means, K-medoids, hierarchical cluster (HC) analysis, expectation–maximization using Gaussian mixture models (EM/GMM), affinity propagation, and mini-batch K-means, to find cluster pattern of landslides, which then acts as training data for the landslide detection problem.

Deep learning-based approaches. The convolutional neural network (CNN) is the most common technique for these approaches. A recent study on landslide identification [10] showed that CNN outperforms most machine learning-based approaches such as RF, LR, and SVM. Bui et al. [17] proposed a system combining CNN for classification tasks

and a transformation algorithm Hue - Bi-dimensional empirical mode decomposition (H-BEMD) to locate the landslide region and size. Interestingly, this study ferreted out that the landslide size depends on time. Another approach proposed in [18] combined CNN and a region-growing algorithm for two main tasks: detecting and classifying the landslide, which reached 97% in terms of F1 score. On other hand, time series data was also utilized to detect the structure changes [19], where non-contribute areas including vegetation, water, and buildings were removed from the pre-landslide and post-landslide images, followed up by a CNN model to detect the changes in image patches.

B. LANDSLIDE SEGMENTATION

For landslide problems, semantic segmentation is a prevalent task. Several studies employed U-net [20], which is the state-of-art deep learning model for semantic segmentation tasks, as the main method for landslide detection and segmentation. Landslide detection in the Himalayas from satellite images [21] compared the performance of U-net and common machine learning-based approaches on two popular datasets, including five optical bands from the RapidEye satellite images and ALOS-PALSAR derived topographical data. To evaluate the efficiency of the generalization model for various datasets, a survey of rainfall-induced landslides in Brazil [22] was performed on three datasets including RapidEye satellite images, Normalized Vegetation Index (NDVI), and a digital elevation model (DEM) figured out that the large patch size has better perform in detect landslides in areas similar to the training area and the small patch has more efficiency in landslide detection in areas with different environmental aspects. Li et al. [23] designed a two-phase framework: F-RCNN for detection and U-net for segmentation of landslide from satellite's images, where skip connection is deployed to replace inception block at the second phase of U-net architecture. Moreover, the authors in [24] proposed another method that combines MobileNetV2 and PSPnet to accelerate the speed and reduce the number of parameters, which reduced the misclassification errors and separated the objects more precisely.

C. FEW-SHOT LEARNING

Few-Shot learning (FSL) is a special case of meta-learning [25], which aims to train a model that can perform well with unseen data using a few samples under related tasks. The main idea of FSL is to determine a hypothesis space of hypotheses and estimates an optimal hypothesis. FSL included interesting variants such as one-short learning [26]–[28] that classifies each label with only one sample for each class and zero-shot learning [29]–[32] that deals with unseen data using data description without any labeled samples.

FSL has been widely adopted in classification, object detection, and recognition. For example, the authors in [33], [34] showed that FSL achieved better performance in hyperspectral images (HSI) classification, which typically requires hundreds or thousands of labeled samples. Following up, Liu et al. proposed deep few-shot learning for hyperspectral image classification [35], in which the crucial concept is to exploit the training dataset to provide a metric space that can generalize to the classes in the unseen dataset. The suggested approach achieved a higher classification accuracy than the traditional semi-supervised methods by evaluating four popular HSI data sets. Additionally, in object detection tasks, a class-imbalanced scenario was considered for road object detection using FSL [36], which demonstrated the application of few-shot learning approaches in real-world images under a driving context. Furthermore, in the recognition task, Das et al. proposed a two-stage approach based on few-shot learning for image recognition [37]. In the first-training stage, the authors captured the structure of the data and obtained an embedding space while also predicting the variance of each class. In the second-training stage, the proposed method learned to map the mean-sample representation to class prototype representation in the embedding space.

Few-shot segmentation [38]–[40] is a sub-field of few-shot learning, which utilized FSL in the segmentation tasks and was paid considerable attention in recent years. An early prototype learning for few-shot segmentation was proposed by Nanqing Dong [38]. In that article, the authors introduced a framework based on prototype learning and metric learning that significantly outperformed the baselines on PASCAL VOC 2012 dataset. Wang et al. proposed a novel prototype alignment network, termed PANet, that can effectively utilize the support set's data [39]. Interestingly, PANet introduced the prototype alignment regularization between the support and query by providing a few-shot segmentation reversely from query to support. Most recently, a method of FSL without meta-learning was proposed in [40], which adopted only transductive inference technique for a given query image while taking advantage of the statistics of its unlabeled pixels by maximizing a new loss containing three complementary terms, including the 1) cross-entropy obtained from the labeled support pixels, 2) Shannon entropy of the posteriors on the unlabeled query-image pixels, and 3) a global KL-divergence regularizer. The method achieved competitive performances in the 1-shot learning setting and noticeably improved performance in the 5- and 10-shot scenarios by 5% and 6%, respectively, in comparison with state-of-the-art episodic training approaches.

III. PRE-PROCESSING LAYER

In this section, we describe the data pre-processing step after obtaining images from UAVs/drones. A landslide is a dangerous natural phenomenon that occurs during heavy rain and floods and can cause a lot of damage to people and infrastructure. Furthermore, the landslide may block the whole road, causing traffic circulation obstruction. Due to difficulty in traffic conditions, terrain, and shortage of equipment such as UAVs and drones, it is challenging to collect data and thus the amount of collected data every year is very few. Since deep learning models typically require a huge amount of data to perform well, we need to generate more data to

TABLE 1: Statistics of the number of road images by background scene.

Background scene	Number of images
Forest only	290
Forest mountain	116
Rock mountain	361

TABLE 2: Statistics of the number of real-life landslide images by type.

Types of landslide	Number of images
Rock fall	21
Mud slide	29
Earth flow	66
Depression	33

compensate for a limited number of collected images.

To this end, we build a newly generated dataset via three following steps, namely data crawling, data generation, and data annotation. In the following, we present the three steps for generating landslide images and then building a landslide dataset including generated images and their annotations.

A. DATA CRAWLING

We collect videos of road data recorded by UAV and drone, in several mountain and forest areas in Vietnam. Frames are extracted from videos and we obtain a dataset consisting of 767 road images. For landslide data, we collect 149 images from the internet by manually selecting landslide images that are taken from a high-ground position, which simulates the actual deployment situation when the UAVs/drones take pictures during surveillance. Then, we also annotate the images with many types of slides such as rock falls, mudslides, earth flow, and depression. The data statistics are shown in Tables 1 and 2.

B. DATA GENERATION AND AUGMENTATION

In this section, we explain the procedure to generate images containing landslides that can be utilized to train machine learning/deep learning models. Each synthetically generated image contains a road augmented with a type of landslide listed in Table 3.

To this end, we first need to determine the centerline of the road in road images and the landslide region in landslide images. For extracting the centerline, we adopt the pretrained RoadNet++ model [41] and then apply post-processing to the model output, resulting in a binary image with pixel value 1 corresponding to the centerline and 0 otherwise. For the landslide region, we crop out the region corresponding to the landslide from four types of landslide images using Labelme application.[2] Finally, we randomly insert the landslide region on the centerline of the road in the image, using the Seamless Cloning algorithm [42] method. The region is blended into the road image, making it more realistic. This above-mentioned procedure is illustrated in Fig. 2a. After

[2]Labelme image polygonal annotation tool: https://github.com/wkentaro/labelme.git

TABLE 3: Class information in the LandslidePTIT dataset.

Class	Id	Number of images
Background	0	2963
Road	1	2963
Earth flow	2	1190
Depression	3	596
Mud slide	4	523
Rock fall	5	379

conducting the data generation phase, from 767 and 149 collected road and landslide images, respectively, we obtain a generated dataset consisting of 2963 images, termed the LandslidePTIT dataset.

After a landslide occurs, light rain is typically observed in the surrounding area. Besides, the landslides tend to occur in mountainous areas, thus the images are often captured in foggy weather. To simulate these two weather conditions, we further augment fog and rain to generate images in the LandslidePTIT dataset, which divides the dataset into three parts normal, fog, and rain.[3]

C. DATA ANNOTATION

The images are labeled pixel-wise, i.e., each pixel of an image is annotated with a range of values from 0 to 255 that indicates the class number. For example, in Fig. 2c, id 0, 1, and 2 are background, road, and earth flow, respectively. Table 3 presents the list of classes deployed in our study. The annotations are JSON files that store coordinates of the polygon forming the landslide area.

IV. CLOUD LAYER

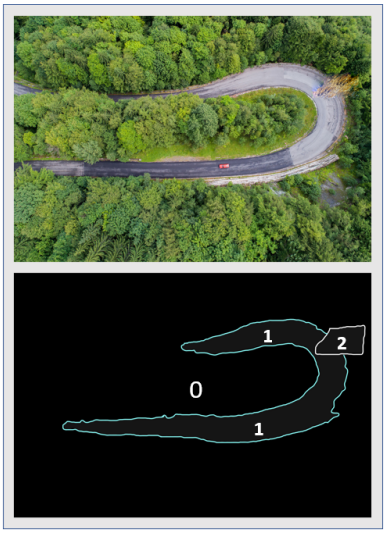
In this layer, we design a model to tackle the landslide detection problem. Specifically, we need to answer the following questions. First, does a landslide appear in a given input image? Second, if a landslide occurs, then does it block the road and may obstruct traffic? Third, what is the class of landslides? Intuitively, if we know the current state of the landslide and the type of landslide, e.g., rockfall, mudslide, etc., then we can issue warnings and take appropriate action based on the situation. To answer these questions, we not only solve the classification task, which classifies an image as the type of having landslide or not but also need to locate the area on the road where the landslide takes place, which is equivalent to the segmentation problem.

Besides, since collecting landslide data is challenging, it is difficult to acquire a large enough dataset to train on conventional deep learning models. Additionally, in reality, each region has different characteristics and there are many types of landslides other than those presented in our LandslidePTIT dataset. For example, in the northwest mountainous areas of Vietnam, the terrain is mainly high mountain forests. In this type of terrain, flash floods often occur and can be considered a new landslide type that we need to detect.

[3]In this study, we generate fog using FoHIS (https://github.com/noahzn/FoHIS) and generate rain using monodepth2 (https://github.com/nianticlabs/monodepth2).

(a) Landslide image generation process.

(b) Fog generation process for data annotation.



(c) Generated landslide image and its corresponding annotation.

FIGURE 2: Examples of data generation and annotation steps in the pre-processing layer.

Therefore, we propose a few-shot segmentation framework to tackle the landslide detection problem, in which we also newly design a cross-feature attentive squeeze network architecture that is customized for the LandslidePTIT dataset.

A. PROBLEM FORMULATION

Few-shot learning aims at performing tasks with only a few labeled data. In our study, we are given two image sets: 1) base set, denoted as $\mathcal{D}^T$ (with base classes), and 2) novel set, $\mathcal{D}^E$ (with novel classes). We note that the novel set also includes road and landslide images, however, the classes in the novel set are different from those in the base set. Specifically, in the training process, the LandslidePTIT dataset is divided into two parts as train set and test set, which are $\mathcal{D}^T$ and $\mathcal{D}^E$ respectively. We denote $\mathcal{C}^T$ and $\mathcal{C}^E$ as the class sets of $\mathcal{D}^T$ and $\mathcal{D}^E$, respectively. In the specific case of the LandslidePTIT dataset, we have $|\mathcal{C}^T| + |\mathcal{C}^E| = 6$.

To detect landslides under the few-shot setting, we follow the idea of episodic learning [43], which is one of the most well-known approaches in the field. Specifically, multiple few-shot tasks are created in the training process, each takes several data samples randomly drawn from the train set $\mathcal{D}^T$ and divides the samples into two sets including a support set, denoted as $\mathcal{S}^T$, and a query set, denoted as $\mathcal{Q}^T$. The model performs the classification and segmentation of the data in the query set based on the data and labels information in the support set. For each few-shot task in the training process, the support set $\mathcal{S}^T$ is a set that has information about the landslide and the corresponding labels while the query set $\mathcal{Q}^T$ includes images that are not labeled, i.e., we need to classify and segment roads and landslides from images in $\mathcal{Q}^T$.

After training completely, we evaluate the model by using the test set $\mathcal{D}^E$ which is also the novel set. The support set $\mathcal{S}^E$ and the query set $\mathcal{Q}^E$ are taken from the test set as the way performed in the training process. We have to use a few-shot learning model to predict the labels of data samples in the query set. Also, during the inference process, the query set is the set of images obtained from the UAVs/drones in the field, which do not have label data and we need to predict the type of landslide and its landslide segmentation. The support set includes several images with a new type of landslide and their labels.

In more detail, we create episodes containing two following sets of samples.

- $\mathcal{S}^* = \left\{\left(x_i^s, a_i^s\right) \mid x_i^s \in \mathbb{R}^{H\times W\times C}, a_i^s \in \mathbb{R}^{H\times W}\right\}_{i=1}^{NK}$, where x_i^s and a_i^s represent a support image with its label from $\mathcal{S}^*$, the superscript $*$ represents T and E for train and test sets, respectively, N is the number of classes in support set, and each class contains K labeled instances, i.e., the so-called N-way K-shot problem. In particular, x_i^s and a_i^s represent a raw image and its corresponding label for a specific category, respectively. Each value in the annotation matrix a_i^s is the class id of the corresponding pixel in image x_i^s. We also denote H, W, C as height, width, and channels of the image x_i^s, respectively.
- $\mathcal{Q}^* = \left\{x_j^q \mid x_j^q \in \mathbb{R}^{H\times W\times C}\right\}_{j=1}^{M}$, where x_j^q is a query

landslide image and M is a number of images in $\mathcal{Q}^*$ that needed to predict. The superscript $*$ represents T and E for train and test sets, respectively, and j indicates the j-th data samples in $\mathcal{Q}^*$.

For classification task, we aim to identify the multi-hot class occurrence vector $\mathbf{y_C} \in \mathbb{R}^N$ via a function $f_\mathbf{C}$; and for segmentation task, we predict the segmentation mask $\mathbf{Y_S} \in \mathbb{R}^{H\times W}$ corresponding to the classes via another function $f_\mathbf{S}$. The two objectives are expressed as follows:

$$\begin{aligned}\mathbf{y_C} &= f_\mathbf{C}(x_j^q, \mathcal{S}^*; \theta_\mathbf{C}),\\ \mathbf{Y_S} &= f_\mathbf{S}(x_j^q, \mathcal{S}^*; \theta_\mathbf{S}),\end{aligned} \tag{1}$$

where $\theta_\mathbf{C}, \theta_\mathbf{S}$ are the learnable parameter of the classification model and the segmentation model, respectively.

In this study, instead of optimizing two functions $f_\mathbf{C}$ and $f_\mathbf{S}$ in (1) separately, we aim at jointly finding a function $f_\mathbf{CS}$ that combines and generalizes two tasks, including few-shot classification and segmentation (FS-CS). It can predict multi-label background-aware class occurrences and also segmentation maps. The integrative FS-CS model $f_\mathbf{CS}$ (with learnable parameter $\theta_\mathbf{CS}$) take as input query image x_j^q and support set $\mathcal{S}^*$,

$$\left\{\hat{\mathbf{y}}_\mathbf{C}, \hat{\mathbf{Y}}_\mathbf{S}\right\} = f_\mathbf{CS}(x_j^q, \mathcal{S}^*; \theta_\mathbf{CS}), \tag{2}$$

where $\hat{\mathbf{y}}_\mathbf{C} \in \mathbb{R}^N$ is the multi-hot class occurrence vector and $\hat{\mathbf{Y}}_\mathbf{S} \in \mathbb{R}^{H\times W}$ is the class-wise segmentation mask.

We note that FS-CS is more general than few-shot classification (FS-C) and also exhibits two major advantages over both FS-C and few-shot segmentation (FS-S) as follows:

- FS-CS can classify the query images, which are belonging to none or multiple target classes (i.e., the query is classified into a background class - none if none of the target classes were detected). Therefore, in a real-life use case, if a landslide does not occur, then the system still operates properly without any warnings.
- FS-CS relaxes the assumption such that the query class set can be a subset of the support class set while the conventional FS-S [39], [44], [45] assumes the query class set exactly matches the support class set.

To solve (2), we need to extract N probability maps corresponding to each class in the support set, which is typically referred to as the class-wise foreground map set, $\mathcal{Y}$, comprised of $\mathbf{Y}^{(n)} \in \mathbb{R}^{H\times W}$ for N classes. Where $\mathbf{Y}^{(n)}$ is the probability map of a class (each position on the map represents the probability of the position being on a foreground region of the corresponding class) and has the same size as the input image $H \times W$. We have:

$$\mathcal{Y} = f(x_j^q, \mathcal{S}^*; \theta) = \left\{\mathbf{Y}^{(n)}\right\}_{n=1}^{N}, \tag{3}$$

where f is the model before the post-processing step and θ is the learnable parameter of the model. $\mathcal{Y}$ is then post-processed to extract $\hat{\mathbf{y}}_\mathbf{C}$ and $\hat{\mathbf{Y}}_\mathbf{S}$ (see Section IV.C.1 for further details).

B. MODEL ARCHITECTURE

To solve (3), we propose a new model architecture, termed CF-ASNet, built upon the state-of-the-art ASNet model [46]. The overall procedure of CF-ASNet is presented in Fig. 3. As illustrated in the figure, we first extract feature maps of a query image (depicted in red) and a support image (depicted in green) from a backbone network, which is illustrated by a trapezoid shape.[4] In this backbone network, three features of an image are extracted from the three last blocks, i.e., blocks 2, 3, and 4 in Fig. 3. Each feature maps pairs with the same level are then used to construct hypercorrelations - the first pyramidal correlation box in the figure. Secondly, the model then learns to transform the correlation through an attentive squeeze block whose details are presented in Fig. 4a by gradually squeezing the support dimension on each query dimension, yielding the high-level hypercorrelations that are later employed to produce the mask prediction map. Finally, in the producing process, two adjacent correlations are cross featured using a network termed the cross feature layer. Each high-level correlation tensor pair after processing results in a feature map, is upsampled and combined with the same query dimension size correlation using broadcast and squeeze layers, whose details are described in Sections IV-B3 and IV-B4, respectively. The earliest feature map is fed to a convolutional decoder, which consists of bi-linear upsampling and interleaved 2D convolution that map the number of dimensional channels to 2 (including foreground and background) and the output spatial size to the input query image size. The detailed implementations are described in the following subsections.

1) Hypercorrelation Construction

Following [45], we construct hypercorrelations between the query image and the support image. First, we extract the features of each image from a pre-trained backbone network (the pre-trained is frozen during the training process). We denote $\mathbf{F}_q^{(b)} \in \mathbb{R}^{C^{(b)}\times H_q^{(b)}\times W_q^{(b)}}, \mathbf{F}_s^{(b)} \in \mathbb{R}^{C^{(b)}\times H_s^{(b)}\times W_s^{(b)}}$ are feature maps of the query image and the support image from block b of the backbone model ($H_*^{(b)}, W_*^{(b)}$ are the feature map size and $C^{(b)}$ is the number of channels). In our work, we adopt ResNet50 and ResNet101 pre-trained as feature extractors, with the output channel sizes of four blocks set to {3, 4, 6, 3} and {3, 4, 23, 3}, respectively. The feature maps pairs of query and support image from block 2, block 3 and block 4 - $\{(\mathbf{F}_q^{(b)}, \mathbf{F}_s^{(b)})\}_{b=2}^4$ are then used to compute cosine similarity as follow:

$$\mathfrak{C}^{(b)}(\mathbf{p}_q, \mathbf{p}_s) = \mathrm{ReLU}\left(\frac{\mathbf{F}_q^{(b)}(\mathbf{p}_q)\cdot\mathbf{F}_s^{(b)}(\mathbf{p}_s)}{\left\|\mathbf{F}_q^{(b)}(\mathbf{p}_q)\right\|\left\|\mathbf{F}_s^{(b)}(\mathbf{p}_s)\right\|}\right), \tag{4}$$

$$\mathbf{p}_q \in [H_q^{(b)}] \times [W_q^{(b)}], \mathbf{p}_s \in [H_s^{(b)}] \times [W_s^{(b)}],$$

[4] In this paper, we adopt ResNet50 as the backbone network.

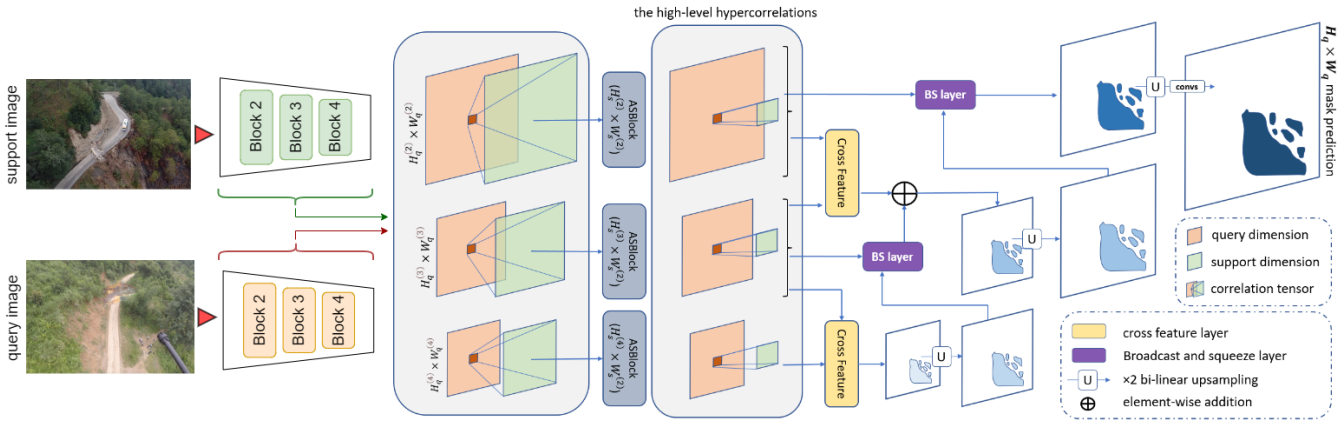


FIGURE 3: **Overview of the proposed Cross Feature and Attentive Squeeze Network (CF-ASNet)**

where $\mathfrak{C}$ is a hypercorrelation and $\mathbf{p}$ is denoted as a matrix position hereafter. Finally, we have a hypercorrelation pyramid: $\{\mathfrak{C}^{(b)} \mid \mathfrak{C}^{(b)} \in \mathbb{R}^{H_q^{(b)} \times W_q^{(b)} \times H_s^{(b)} \times W_s^{(b)} \times C^{(b)}}\}_{b=2}^{4}$.

2) Attentive Squeeze Block

The AS blocks consist of AS layers introduced in [46]. As illustrated in Fig. 4a, the AS blocks transform each correlation tensor $\mathfrak{C} \in \mathbb{R}^{H_q \times W_q \times H_s \times W_s \times C_{\text{in}}}$ to tensors with fixed support dimension size $H_s^{'} \times W_s^{'}$, where C_{in} and C_{out} denote the number of channels of the input and output tensors, respectively, $H_s^{'} \leq H_s$, and $W_s^{'} \leq W_s$. We can consider $\mathfrak{C}$ as a block matrix with size $H_q \times W_q$, Each element of this block matrix, which is called as a support correlation tensor, corresponds to a correlation tensor between each query position $\mathbf{p}_q \in [H_q] \times [W_q]$ and every support position.

In Fig. 4, the rearrange tensor operator expresses the transformation between the correlation tensor and block matrices. Each support correlation tensor is then fed to AS layers to analyze the global context. Finally, after rearranging, we have correlation tensors with a reduced support dimension while the query dimension is preserved, which is called high-level correlations, denoted as $\ddot{\mathfrak{C}}$.

3) Cross Feature Layer

In our study, via empirical experiments, we find that in many cases, the predicted road segmentation encroaches into the ground truth segmentation of the landslide. This observation is due to the fact that the landslide area is quite small compared to the road area and the whole image. Therefore, we propose cross feature layers (CF layers) between two adjacent correlations in a high-level correlation pyramid to enhance the model's ability to segment small objects.

In more detail, CF layers take as input two high-level correlation tensors $\ddot{\mathfrak{C}}^{(b-1)}, \ddot{\mathfrak{C}}^{(b)}$ $(b = 3, 4)$ with size are $H_q \times W_q \times H_s^{'} \times W_s^{'} \times C_{out}$ and $\lceil \frac{H_q}{2} \rceil \times \lceil \frac{W_q}{2} \rceil \times H_s^{'} \times W_s^{'} \times C_{out}$, respectively. First, we rearrange the bigger tensor $\ddot{\mathfrak{C}}^{(b-1)}$ as a matrix block of size $H_s^{'} \times W_s^{'}$ with each elements size $H_q \times W_q \times C_{out}$. The elements then go through convolution layers for downsizing to the query dimension of the smaller tensor $\ddot{\mathfrak{C}}^{(b)}$. Then, the result is rearranged and combined with the smaller tensor in the ratio α, which shall be empirically determined via experiments. The mixed representation is rearranged and then fed to two sequential AS layers until it becomes a point feature of size 1×1. The detailed architecture of CF layers is illustrated in Fig. 4b.

4) Broadcast and Squeeze Layer

For each high-level correlation $\ddot{\mathfrak{C}}^{(b)}$ $(b = 3, 4)$, after processing through CF layers, we have a feature map with the same size as the query dimension of the correlation $\ddot{\mathfrak{C}}^{(b)}$. We then bi-linear upsampling that maps to the size of query dimension of $\ddot{\mathfrak{C}}^{(b-1)}$. Next, the resulted map and the correlation $\ddot{\mathfrak{C}}^{(b-1)}$ are input to the broadcast and squeeze layer (BS layer). The layer first uses broadcasted element-wise addition operator to combine the two inputs, then rearrange the resultant and finally feed the results to two sequential AS layers until the output becomes a point feature of size 1×1. Fig. 4c illustrates the detailed architecture of BS layers.

C. TRAINING PROCEDURE

1) Prediction

After obtaining the set of class-wise foreground maps, $\mathcal{Y}$, we perform the prediction/inference step to archive the multi-hot class occurrence, $\hat{\mathbf{y}}_{\mathbf{C}}$ and the segmentation mask, $\hat{\mathbf{Y}}_{\mathbf{S}}$.

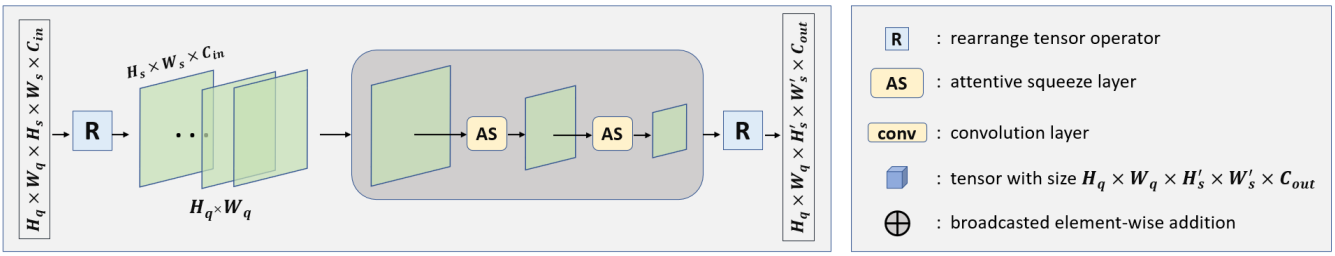
For classification. With each class probability map $\mathbf{Y}^{(n)}$, if the maximum value of the matrix is greater than a predefined threshold δ, then it means that the object with class n is in the query image:

$$\hat{\mathbf{y}}_{\mathbf{C}}^{(n)} = \begin{cases} 1 \text{ if } \max_{\mathbf{p} \in [H] \times [W]} \mathbf{Y}^{(n)}(\mathbf{p}) \geq \delta \\ 0 \text{ otherwise,} \end{cases} \tag{5}$$

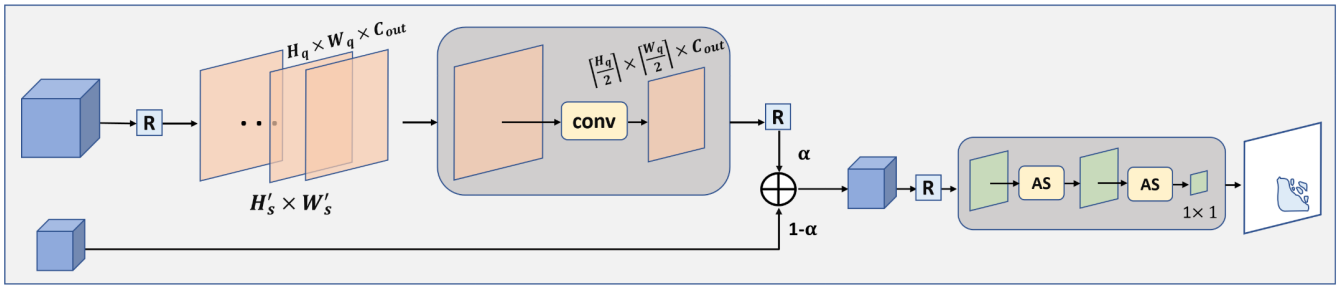
where $\mathbf{p}$ represents the position in the matrix.

For segmentation. We compute the final segmentation mask $\hat{\mathbf{Y}}_{\mathbf{S}}$ by choosing the class that has the highest probability, for each pixel position:

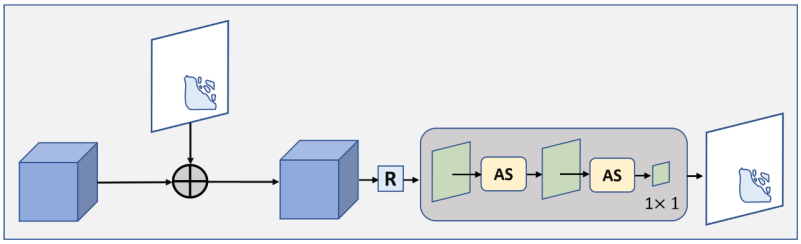
$$\hat{\mathbf{Y}}_{\mathbf{S}}(\mathbf{p}) = \underset{n \in [N+1]}{\arg\max} \mathbf{Y}^{(n)}, \tag{6}$$



(a) The attentive squeeze block (AS block). Here, the legend placed at the right side is universal for all Figs. 4a, 4b, and 4c.



(b) Cross feature layer (CF layer).



(c) Broadcast and squeeze layer (BS layer).

FIGURE 4: The architecture of the attentive squeeze block, cross feature layer, and broadcast and squeeze layer.

where $\mathbf{Y}^{(N+1)}$ is the background probability map derived from the class-wise foreground maps, expressed as follows:

$$\mathbf{Y}^{(N+1)} = \frac{1}{N} \sum_{n=1}^{N} \left(\mathbf{1} - \mathbf{Y}^{(n)} \right). \tag{7}$$

2) Loss Function

FS-CS learner use segmentation loss in training, which is formulated as the average cross-entropy between the class distribution at each individual position and its ground-truth segmentation annotation:

$$\mathcal{L} = -\frac{1}{(N+1)} \frac{1}{HW} \sum_{n=1}^{N+1} \sum_{\mathbf{p} \in [H] \times [W]} \mathbf{Y}_{\text{gt}}^{(n)}(\mathbf{p}) \log \mathbf{Y}^{(n)}(\mathbf{p}), \tag{8}$$

where $\mathbf{Y}_{\text{gt}}$ denotes the ground-truth segmentation mask.

V. EXPERIMENTAL RESULTS

A. BASELINES AND PERFORMANCE METRICS

We compare our proposed method with four state-of-the-art approaches:

- **Path Aggregation Network (PANet) [39]**: PANet reaches 1st place in the COCO 2017 Challenge Instance Segmentation task and 2nd place in the Object Detection task without large-batch training.
- **Prior Guided Feature Enrichment Network (PFENet) [44]** is one of state-of-the-art FSL methods.
- **Hypercorrelation Squeeze Networks (HSNet) [45]** is a novel framework for FS-CS that analyzes complex feature correlations in a fully-convolutional manner using light-weight 4D convolutions.
- **Attentive Squeeze Network (ASNet) [46]**: This model achieves state-of-the-art performance on both FS-CS and FS-S tasks.

For multi-label classification evaluation metrics, we use the $0/1$ exact ratio: $\text{ER} = \mathbb{1}[\hat{\mathbf{y}}_{\mathbf{C}} = \mathbf{y}_{\mathbf{gt}}]$, where $\mathbf{y}_{\text{gt}}$ is the ground truth multi-hot class occurrence vector and $\hat{\mathbf{y}}_{\mathbf{C}}$ is predicted vector of the model. For segmentation, we use mean IoU: $\text{mIoU} = \frac{1}{N} \sum_n \text{IoU}_n$, where IoU_n denotes an IoU [47] value of n_{th} class.

B. EXPERIMENTAL SETUP

We choose ResNet50 and ResNet101 trained on ImageNet as our backbone networks for comparison with other methods. The CF-ASNet is trained using the Adam optimizer [48] with a learning rate of 10^{-4} for label segmentation. We train the

model with two cases, 1-way 1-shot, and 2-way 1-shot.

C. EXPERIMENTAL RESULTS

In this section, we are interested in the following five research questions and we design experiments to investigate the answers to these questions.

Q1. How do different data augmentation methods affect performance?
Q2. How much does CF-ASNet improve the performance on the LandslidePTIT dataset in comparison with other baselines?
Q3. What is the performance of CF-ASNet and baselines with respect to the number of ways?
Q4. How do the hyperparameters α and δ affect the performance of the CF-ASNet model?
Q5. How close is the prediction of the CF-ASNet model to the ground truth?

1) Comparison of two data generation methods (Q1):

To validate the effectiveness of the proposed data generation method presented in Section III, we first train our model based on two datasets created by two variants of the data generation procedure as follows:

- (1) *LandslidePTIT dataset:* We insert landslide as described in Section III.
- (2) *Dataset Landslide-Normal (LN):* We insert the landslide on the designated location of the image containing roads.

Then, we evaluate the performance of all methods on another real-world data from [49], termed Landslide-Premise, which consists of 400 real-world landslide images captured by UAVs. That is, Landslide-Premise dataset acts as the test data. Table 4 shows the experimental results of all few-shot models, including CF-ASNet and competitive baselines trained on the two datasets that we created.

- Performance of all models trained using *LandslidePTIT dataset* are higher than that when training with *LN* in both classification and data segmentation tasks. Therefore, the use of blending will affect the accuracy of the data classification model.
- We illustrate the qualitative result of two datasets in Fig. 5. As shown in Fig. 5a, the landslide overwritten on the original image using (1) is blended to make the data smoother and more realistic. As for (2), the included landslide in Fig. 5b looks coarser and does not look like a normal landslide.

Overall, we can see that the proposed data generation method achieves better results on actual data, showing the applicability of using this synthetic dataset as training data.

2) Performance comparison of CF-ASNet and other baselines (Q2):

From this part onwards, we use the LandslidePTIT dataset for both training and evaluation as this dataset is larger and more diverse than the Landslide-Premise dataset, making it more reasonable to evaluate competitive models. In this part, we divide LandslidePTIT into train and test sets. The train set consists of 1893 images that belong to 4 classes including Rockfall, Mudslide, Earth flow, and Road, in which the backgrounds of Roads are Forest only and Forest mountain. The test set contains 1070 images with two classes Depression and Road with Rock mountain background, respectively.

(a) An image from LandslidePTIT

(b) An image from LN

FIGURE 5: Visual comparison between the two generation methods.

Table 5 presents the performance for both classification and segmentation tasks of all competitive schemes, including PANet, PFENet, HSNet, ASNet, and our proposed method CF-ASNet. We discuss interesting points as follows.

- We find that our CF-ASNet model achieves the best performance in terms of both 0/1 ER and mIoU among all the few-shot learning models in the 1-way 1-shot setting. Notably, the performance of CF-ASNet achieves 1% higher ER and 0.5% higher mIoU than the results obtained by ASNet, accounting for the contribution of the new cross-feature layer.
- The performance of CF-ASNet is much higher than those of PANet and HSNet, with the gain from 10 to 15% with the classification metric 0/1 ER, and from 3 to 7% with the segmentation metric mIoU. This is due to the well-designed architecture including the combination of ASNet and hypercorrelation computation.

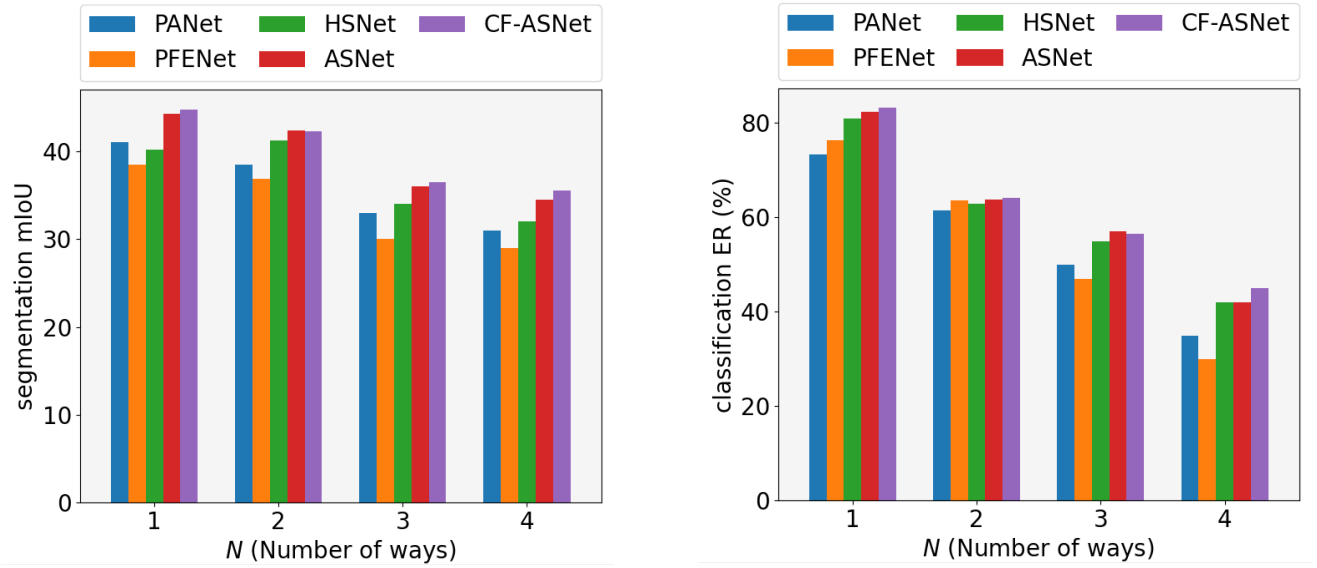
3) Comparison between all competitive schemes with respect to the number of ways (Q3):

In Fig. 6, we perform an experiment where the number of ways is adjustable. In the case when the number of ways is 1, we randomly choose a type of landslide to put in the test set. When the number of ways $N > 1$, i.e., $N \in \{2, 3, 4\}$, we randomly divide the classes of the training set to include $N - 1$ landslide labels, which means the test set includes $4 - (N - 1)$ other labels of landslide.

As shown in Fig. 6a, in the case of 1-way 1-shot, our CF-ASNet method works best with an accuracy of 81%. Besides, we also notice a decrease in both ER and mIoU as we increase the number of ways from 1 to 4. Overall, the performance of CF-ASNet is always the highest, except for

TABLE 4: Results on few-shot learning of the proposed models

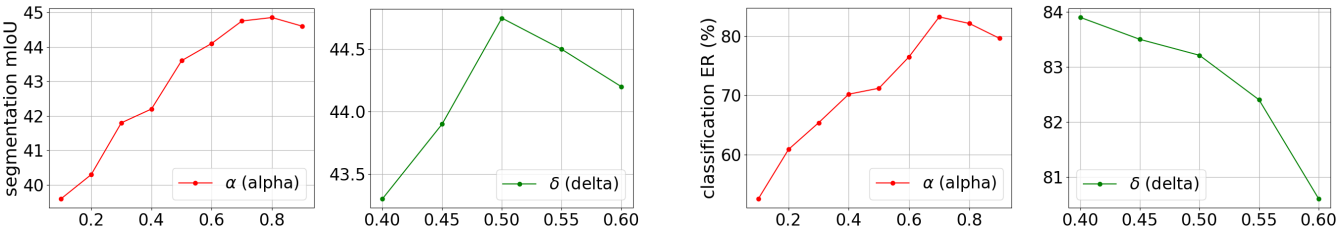
	Dataset LB				Dataset LN			
method	1-way 1-shot		2-way 1-shot		1-way 1-shot		2-way 1-shot	
metric	cls. 0/1 ER	seg. mIoU	cls. 0/1 ER	seg. mIoU	cls. 0/1 ER	seg. mIoU	cls. 0/1 ER	seg. mIoU
PANet [39]	70. 6	31.0	62.2	32.7	63.36	23.6	53.4	25.1
PFENet [44]	75.9	35.1	61.7	32.2	62.8	27.3	46.5	24.9
HSNet [45]	78.5	36.2	64.5	**34.6**	64.0	28.8	**62.1**	28.0
ASNet [46]	80.4	**38.5**	64.2	33.4	66.7	31.1	61.7	**28.5**
CF-ASNet	**81.4**	**38.5**	**65.3**	34.1	**68.8**	**31.5**	61.9	27.7



(a) Segmentation (mIoU)

(b) Classification (exact ratio)

FIGURE 6: N-way 1-shot FS-CS performance comparison of five methods by varying N from 1 to 4 on LandslidePTIT



(a) Segmentation (mIoU)

(b) Classification (exact ratio)

FIGURE 7: Performance of 1-way 1-shot CF-ASNet on LandslidePTIT when varying α and δ

TABLE 5: Performance comparison of CF-ASNet and others on our LandslidePTIT dataset

Performance results				
methods	1-way 1-shot		2-way 1-shot	
metric	cls. 0/1 ER	seg. mIoU	cls. 0/1 ER	seg. mIoU
PANet	73.36	40.99	61.5	38.42
PFENet	76.45	38.54	63.55	36.85
HSNet	80.94	40.18	62.9	41.26
ASNet	82.43	44.29	63.74	**42.42**
CF-ASNET	**83.21**	**44.75**	**64.11**	42.3

the case where N is set to 3, where the performance of ASNet is slightly superior.

In Fig. 6b, we illustrate the superiority of CF-ASNet for the segmentation task when varying the number of ways.

4) Hyperparameter Sensitivity Analysis (Q4):

In Fig. 7, we present the sensitivity analysis of the parameters α of the Cross Feature Layer and threshold δ of the prediction. When analyzing α, we set the default value of δ as 0.5 in all experiments.

When we increase α from 0.1 to 0.7, the performance of both segmentation and classification tasks exhibits an upward trend. As shown in Fig. 7a, with α = 0.8, the segmentation

mIoU value reaches the highest value (approximately 45). In Fig. 7b, when we set α to 0.7, the accuracy of the classification model peaked at approximately 82%. In general, we empirically find that when α ranges from 0.7 to 0.8, the model performs well. We note that we can validate the choice of parameter α using various validation methods such as k-fold validation.

Next, we adjust the threshold coefficient δ while the parameter α is set to 0.7. When $\delta < 0.5$, the ER is high and the mIoU is at a lower value. When $\delta > 0.5$, the mIoU experiences a decrease while ER also drops. In our experiment, we set the threshold $\delta = 0.5$.

5) Qualitative Study (Q5):

We perform the qualitative study by showing the results obtained from the CF-ASNet model, in the form of landslide images captured by drones in Fig. 8. The CF-ASNet model takes two sets, including query and support sets, as the input. On the left side of Fig. 8, we illustrate the query sets that are images extracted from LandslidePTIT. In the middle of the figure, we show examples of the support set used in the training process, where red and blue marks represent the pixels containing the landslide and the road, respectively. The prediction result of CF-ASNet model is displayed on the right side of the figure, side-by-side with the ground-truth images. From the example, one can see that the CF-ASNet model provides well-segmented roads and landslides.

VI. CONCLUSION

In this study, we introduced a swift detection system that can locate the occurrence of landslides on roads. Due to the difficulties in data collection of real-world landslide images, the system was trained upon a synthetic dataset, created from our new design procedure. We also proposed a new few-shot segmentation method, termed CF-ASNet, to enhance the capacity of the system in detecting landslides in real-world situations. Experimental results showed promising applicability of the generated dataset as well as the proposed CF-ASNet model in classifying and segmenting damages caused by landslides on roads.

Future avenues include the enrichment of the synthetic dataset using additional real-world conditions and terrains. We also plan to conduct an empirical study when deploying this system in the mountainous areas of Vietnam.

REFERENCES

[1] H. Bourenane and Y. Bouhadad, "Impact of land use changes on landslides occurrence in urban area: the case of the Constantine City (NE Algeria)," Geotechnical Geological Eng., vol. 39, no. 6, pp. 1–21, Apr. 2021.

[2] L. Picarelli, S. Lacasse, and K. K. Ho, "The Impact of Climate Change on Landslide Hazard and Risk," in Proc. Worksh. World Landslide Forum, Kyoto, Japan, Nov. 2020, pp. 131–141.

[3] M. H. Nguyen, T. V. Ho, T. K. Nguyen, and M. D. Do, "Modeling and simulation of the effects of landslide on circulation of transports on the mountain roads," Int. Journal Adv. Comput. Sci. Appl. (IJACSA), vol. 6, no. 8, Sep. 2015.

[4] Z. Ma, G. Mei, and F. Piccialli, "Machine learning for landslides prevention: a survey," Neural Comput. Appl., vol. 33, no. 17, pp. 10 881–10 907, Nov. 2021.

[5] F. S. Tehrani, M. Calvello, Z. Liu, L. Zhang, and S. Lacasse, "Machine learning and landslide studies: Recent advances and applications," Natural Hazards, pp. 1–49, Jun. 2022.

[6] N. Prakash, A. Manconi, and S. Loew, "A new strategy to map landslides with a generalized Convolutional Neural Network," Scientific reports, vol. 11, no. 1, pp. 1–15, Jun. 2021.

[7] O. Ghorbanzadeh, A. Crivellari, P. Ghamisi, H. Shahabi, and T. Blaschke, "A comprehensive transferability evaluation of U-Net and ResU-Net for landslide detection from Sentinel-2 data (case study areas from Taiwan, China, and Japan)," Scientific Reports, vol. 11, no. 1, pp. 1–20, Jul. 2021.

[8] A. K.-F. Lui, Y.-H. Chan, and M.-F. Leung, "Modelling of pedestrian movements near an amenity in walkways of public buildings," in Proc. 8th Int. Conf. Control Automat. Robot. (ICCAR), Beijing, China, April 2022, pp. 394–400.

[9] A. K.-F. Lui, Y.-H. Chan, and L. Man-Fai, "Modelling of destinations for data-driven pedestrian trajectory prediction in public buildings," in Proc. IEEE Int. Conf. Big Data (Big Data), Orlando, FL, USA, Dec. 2021, pp. 1709–1717.

[10] H. Wang, L. Zhang, K. Yin, H. Luo, and J. Li, "Landslide identification using machine learning," Geosci. Frontiers, vol. 12, no. 1, pp. 351–364, Jan. 2021.

[11] M. Marjanovic, B. Bajat, and M. Kovacevic, "Landslide susceptibility assessment with machine learning algorithms," in Proc. Int. Conf. Intell. Netw. Collaborative Syst., Barcelona, Spain, Nov. 2009, pp. 273–278.

[12] W. Chen, X. Li, Y. Wang, G. Chen, and S. Liu, "Forested landslide detection using LiDAR data and the random forest algorithm: A case study of the Three Gorges, China," Remote Sens. Environ., vol. 152, pp. 291–301, Sep. 2014.

[13] C. P. Poudyal, "Landslide susceptibility analysis using decision tree method, Phidim, Eastern Nepal," Bull. Dept. Geol., vol. 15, pp. 69–76, Jan. 2012.

[14] P. Tsangaratos and I. Ilia, "Comparison of a logistic regression and Naïve Bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size," Catena, vol. 145, pp. 164–179, Oct. 2016.

[15] C. Lian, Z. Zeng, W. Yao, and H. Tang, "Extreme learning machine for the displacement prediction of landslide under rainfall and reservoir level," Stochastic Environ. Res. risk assessment, vol. 28, no. 8, pp. 1957–1972, Dec. 2014.

[16] B. Pokharel, O. F. Althuwaynee, A. Aydda, S.-W. Kim, S. Lim, and H.-J. Park, "Spatial clustering and modelling for landslide susceptibility mapping in the north of the Kathmandu Valley, Nepal," Landslides, vol. 18, no. 4, pp. 1403–1419, Nov. 2021.

[17] T.-A. Bui, P.-J. Lee, K.-Y. Lum, C. Loh, and K. Tan, "Deep learning for landslide recognition in satellite architecture," IEEE Access, vol. 8, pp. 143 665–143 678, Aug. 2020.

[18] H. Yu, Y. Ma, L. Wang, Y. Zhai, and X. Wang, "A landslide intelligent detection method based on CNN and RSG_R," in Proc. IEEE Int. Conf. Mechatronics Automat.(ICMA), Takamatsu, Japan, Aug. 2017, pp. 40–44.

[19] A. Ding, Q. Zhang, X. Zhou, and B. Dai, "Automatic recognition of landslide based on CNN and texture change detection," in Proc. 31st Youth Academic Annu. Chinese Assoc. Automat. (YAC), Wuhan, Hubei Province, China, Nov. 2016, pp. 444–448.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in Proc. Int. Conf. Med. image Comput. computer-assisted intervention, Munich, Germany, Oct 2015, pp. 234–241.

[21] S. R. Meena, L. P. Soares, C. H. Grohmann, C. van Westen, K. Bhuyan, R. P. Singh, M. Floris, and F. Catani, "Landslide detection in the Himalayas using machine learning algorithms and U-Net," Landslides, vol. 19, no. 5, pp. 1209–1229, Fer. 2022.

[22] L. P. Soares, H. C. Dias, G. P. B. Garcia, and C. H. Grohmann, "Landslide segmentation with deep learning: Evaluating model generalization in rainfall-induced landslides in Brazil," Remote Sens., vol. 14, no. 9, p. 2237, May 2022.

[23] H. Li, Y. He, Q. Xu, J. Deng, W. Li, and Y. Wei, "Detection and segmentation of loess landslides via satellite images: A two-phase framework," Landslides, vol. 19, no. 3, pp. 673–686, Jan. 2022.

[24] Z. Li and Y. Guo, "Semantic segmentation of landslide images in Nyingchi region based on PSPNet network," in Proc. 7th Int. Conf. Inf. Sci. Control Eng. (ICISCE), Kopaonik, Serbia, Mar. 2020, pp. 1269–1273.

[25] J. Vanschoren, "Meta-learning: A survey," arXiv preprint arXiv:1810.03548, Oct. 2018.

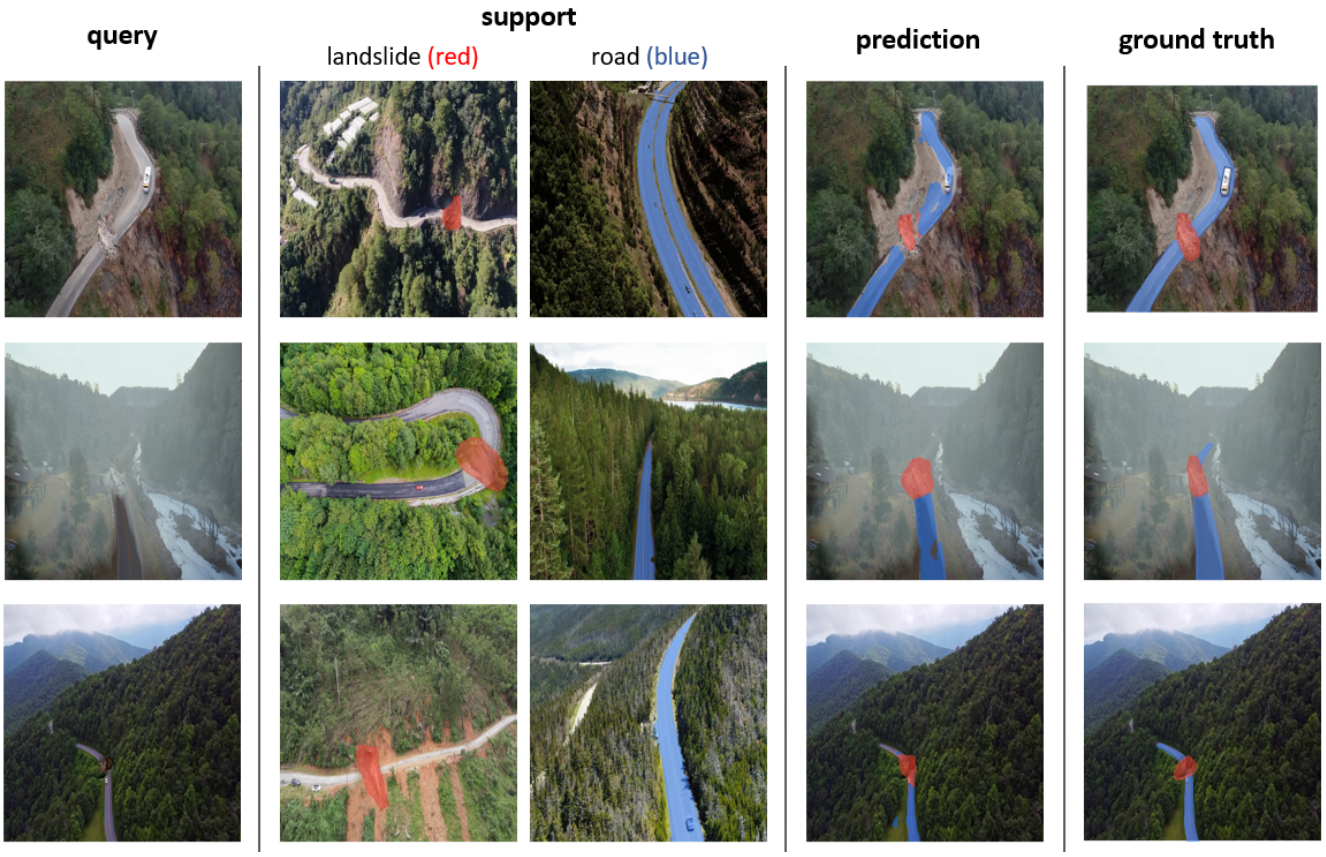


FIGURE 8: The prediction of CF-ASNet model

[26] M. Woodward and C. Finn, "Active one-shot learning," arXiv preprint arXiv:1702.06559, Fer. 2017.

[27] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in Proc. Annu. Meeting Cogn. Sci. Soc., vol. 33, no. 33, Boston,USA, Jul. 2011.

[28] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 4, pp. 594–611, Apr. 2006.

[29] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 3, pp. 453–465, Jul. 2013.

[30] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-data learning of new tasks." in Proc. Assoc. Adv. Artif. Intell. (AAAI), vol. 1, no. 2, Chicago, Illinois, Jul. 2008, p. 3.

[31] M. Rohrbach, M. Stark, and B. Schiele, "Evaluating knowledge transfer and zero-shot learning in a large-scale setting," in Proc. Conf. Comput. Vision Pattern Recognit. (CVPR), Colorado Springs, CO, USA., Jun. 2011, pp. 1641–1648.

[32] Z. Ding, M. Shao, and Y. Fu, "Low-rank embedded ensemble semantic dictionary for zero-shot learning," in Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR), Honolulu, Hawaii, Jul. 2017, pp. 2050–2058.

[33] G. S. Dhillon, P. Chaudhari, A. Ravichandran, and S. Soatto, "A baseline for few-shot image classification," arXiv preprint arXiv:1909.02729, Sep. 2019.

[34] D. Chen, Y. Chen, Y. Li, F. Mao, Y. He, and H. Xue, "Self-supervised learning for few-shot image classification," in Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), Toronto, Canada, Jun. 2021, pp. 1745–1749.

[35] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," IEEE Trans. Geosci. Remote Sens., vol. 57, no. 4, pp. 2290–2304, Oct. 2018.

[36] A. Majee, K. Agrawal, and A. Subramanian, "Few-shot learning for road object detection," in Proc. Assoc. Adv. Artif. Intell.(AAAI) Workshop Meta-Learning MetaDL Challenge, Vancouver Convention Centre, Vancouver, Canada., Fer. 2021, pp. 115–126.

[37] D. Das and C. G. Lee, "A two-stage approach to few-shot learning for image recognition," IEEE Trans. Image Process., vol. 29, pp. 3336–3350, Dec. 2019.

[38] N. Dong and E. P. Xing, "Few-shot semantic segmentation with prototype learning." in Proc. British Mach. Vision Conf. (BMVC), vol. 3, no. 4, Newcastle, UK, Sep. 2018.

[39] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in Proc. IEEE/CVF Int. Conf. Comput. Vision, Long Beach, California, Jun. 2019, pp. 9197–9206.

[40] M. Boudiaf, H. Kervadec, Z. I. Masud, P. Piantanida, I. Ben Ayed, and J. Dolz, "Few-shot segmentation without meta-learning: A good transductive inference is all you need?" in Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR), Nashville, Tennessee Virtual/Online, Jun. 2021, pp. 13 979–13 988.

[41] Y. Liu, J. Yao, X. Lu, M. Xia, X. Wang, and Y. Liu, "RoadNet: Learning to comprehensively analyze road networks in complex urban scenes from high-resolution remotely sensed images," IEEE Trans. Geosci. Remote Sens., vol. 57, no. 4, pp. 2043–2056, Oct. 2018.

[42] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in Assoc. Comput. Machinery's Special Interest Group Comput. Graph. Interactive Techn. (ACM SIGGRAPH) Papers, Jul. 2003, vol. 22, no. 3, pp. 313–318.

[43] A. Li, T. Luo, T. Xiang, W. Huang, and L. Wang, "Few-shot learning with global class representations," in Proc. IEEE/CVF Int. Conf. Comput. Vision (ICCV), Seoul, Korea, Nov. 2019, pp. 9715–9724.

[44] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, "Prior guided feature enrichment network for few-shot segmentation," Comput. Res. Repository (CoRR), vol. abs/2008.01449, Aug. 2020.

[45] J. Min, D. Kang, and M. Cho, "Hypercorrelation squeeze for few-shot segmentation," Comput. Res. Repository (CoRR), vol. abs/2104.01538, Apr. 2021. [Online]. Available: https://arxiv.org/abs/2104.01538

[46] D. Kang and M. Cho, "Integrative Few-Shot Learning for Classification

and Segmentation," in Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit., New Orleans, Louisiana, Jun. 2022, pp. 9979–9990.
[47] S. H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. D. Reid, and S. Savarese, "Generalized intersection over union: A metric and A loss for bounding box regression," Comput. Res. Repository (CoRR), vol. abs/1902.09630, Fer. 2019. [Online]. Available: http://arxiv.org/abs/1902.09630
[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, Dec. 2014.
[49] H. N. Vu, H. M. Nguyen, C. D. Pham, A. D. Tran, K. N. Trong, C. Pham, and V. H. Nguyen, "Landslide detection with unmanned aerial vehicles," in Proc. Int. Conf. Multimedia Anal. Pattern Recognit. (MAPR), Hanoi,Vietnam, Oct. 2021, pp. 1–7.

DAT TRAN-ANH was born in Hanoi, Vietnam, in 1997. He received the B.E. degree in security information and M.S degree in computer science from Posts and Telecommunications Institute of Technology, Hanoi, in 2020 and 2022, respectively. He is currently an AI Engineer with MobiFone Infomation Technology Center. His research interests include image processing, signal preprocessing, and machine learning/deep learning.

BAO BUI-QUOC was born in Thai Binh, Viet Nam in 1998. He received the B.E. degree in applied mathematics and informatics from Hanoi University of Science and Technology, Ha Noi, in 2021. He is currently an AI Engineer at MobiFone Infomation Technology Center. His research interests include image processing, computer vision, and machine learning/reinforcement learning.

ANH VU-DUC was born in Hanoi, Vietnam, in 2000. He is a fourth-year student at Posts and Telecommunications Institute of Technology, Hanoi. His major is Computer Science and his research interests include image processing, computer vision, natural language processing, and machine learning/deep learning

TRUNG-ANH DO received the B.S. and the M.S. degrees in telecommunications and electronics engineering from the Hanoi University of Technology, Hanoi, Vietnam, in 2009 and 2011, respectively, and the Ph.D. degree in telecommunications engineering from the Posts and Telecommunications Institute of Technology, Hanoi. From 2011 to 2014, he was a Researcher with the Research Institute of Posts and Telecommunications, Hanoi, Vietnam. From 2014 to 2016, he was with the Communications & Networking Laboratory, Dankook University, Yongin, South Korea. Since December 2016, he has been with The Department of Science & Technology Management and International Cooperation, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam. His research interests include wireless communications, mobile computing, data mining, and machine learning

VIET-HUNG NGUYEN received his MSc degree in 2009 from Grenoble Institute of Technology and his PhD in Signal Processing and Telecommunications from University of Rennes 1, France. He is currently a Lecturer at Posts and Telecommunications Institute of Technology, Vietnam. His research activities are focus on antennas and microwave circuit design for next-generation wireless communication systems.

HOAI NAM VU was born in Ha Noi, Viet Nam in 1990. He received the B.E. degree in Electronic and Telecommunication Engineering from the Ha Noi University of Science and Technology, Ha Noi, Viet Nam in 2013 and the M.S. degree in Electronic and Computer engineering from Chonnam National University, Gwangju, South Korea, in 2015. He is currently pursuing the Ph.D. degree in Computer Science at Posts and Telecommunications Institute of Technology, Ha Noi. Since 2016, he has been a lecturer with Computer Science Department, Posts and Telecommunications Institute of Technology, Viet Nam. His research interests include sensor signal processing, drone-based image processing, machine learning, and deep learning.

CONG TRAN received his doctoral degree in computer science from Dankook University, Yongin, Republic of Korea, in 2021. He previously received his M.Sc. in computer science in 2014 and his B.Sc. in network and communication in 2009 from Vietnam National University, Hanoi, Vietnam. Since September 2021, he has been with the Faculty of Information Technology, Posts & Telecommunication Institute of Technology, Hanoi, Vietnam, as a lecturer. His research interests include social network analysis, data mining, and machine learning.