https://doi.org/10.1016/j.knosys.2021.107562

This is the Pre-Published Version.

Knowledge-Based Systems 234 (2021) 107562

A robust end-to-end deep learning framework for detecting Martian landforms with arbitrary orientations

Shancheng Jiang [a,*], Fan Wu [b], K.L. Yung [a], Yingqiao Yang [a], W.H. Ip [a], Ming Gao [c,d], James Abbott Foster [a]

[a] *Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, 999077, Hong Kong, P. R. China*
[b] *School of Intelligent Systems Engineering, Sun Yat-Sen University, No. 135, Xingang Xi Road, Guangzhou, 510275, P. R. China*
[c] *School of Management Science and Engineering, Dongbei University of Finance and Economics, No. 217 Jianshan Street, Shahekou District, Dalian, 116025, P. R. China*
[d] *Center for Post-doctoral Studies of Computer Science, Northeastern University, Shenyang, P. R. China*

Keywords:
Deep learning Object detection
Rotated single shot multibox detector Match strategy
Autoencoder

a b s t r a c t

With increasingly massive amounts of high-resolution images of Mars, automated detection of geological landforms on Mars has received widespread interest. It is significant for acquiring knowledge of distant planetary surfaces and processes, or manifold onboard applications such as spacecraft motion estimation and obstacle avoidance. This is a challenging task, not only because of the multiple sizes of targets and complex image backgrounds, but also the various orientations of some bar-shaped landforms in satellite images captured from the top view. The existing methods for directed landform detection require several pre or post-processing operations to extract possible regions of interest and final detection results with orientation, which are very time consuming. In this paper, a new end-to-end deep learning framework is developed for detecting arbitrarily-directed landforms. This framework, named Rotated-SSD (Single Shot MultiBox Detector, SSD), can locate and identify different landforms on Mars in one pass, by using rotatable anchor-box based mechanism. To enhance its robustness against angle variation of the targets and complex backgrounds, a new efficient match strategy is proposed for anchoring default boxes to ground truth boxes in the model training process and an autoencoder-based unsupervised pre-training operation is introduced to improve both the model training and inference performance. The proposed framework is tested for detection of bar- shaped buttes and dark slope streaks on satellite images. The detection results show that our framework can significantly contribute to onboard motion estimation systems. The comparative results demonstrate that the proposed match strategy outperforms other state-of-the-art match strategies with regard to model training efficiency and prediction accuracy. The pre-training strategy can facilitate the training of deep architectures in case of limited available training data.

1. Introduction

As interest in Mars exploration grows in recent decades, the enormous number of images generated by orbiting spacecraft and surface rovers continues to grow on a daily basis. These images are significant materials for the study of Martian geology, including the age of planetary features, physical parameters, the effect of climate and erosion, transposition mechanics, etc. In addition, these data are desired to provide knowledge for per- forming onboard-mode tasks, which means that a spacecraft or a rover can autonomously analyze its collected imagery in real time and take appropriate actions by using the onboard processing system. For example, the descent and landing system of a spacecraft needs to deliver the rover and payload safely onto the Martian surface. This is conducted by an intelligent system, named as the motion estimation system, which estimates velocity and position by processing the imagery, altitude and inertial measurements. Another case is the onboard navigation system embedded on rovers: using their cameras and stereo imagers with associated algorithms, the rovers can figure out for themselves the safest and best way to a certain destination of interest. To fulfill these tasks, an efficient landform detection algorithm is of paramount importance, which can localize and identify some common types of geologic landforms in real time, such as volcanoes, craters, cones, dunes, dark slope steaks, etc.

Based on retrospective study on remote sensing data analysis, the importance of object detection for target landforms has been highlighted [1–5]. Most previous research on landform detection have focused on the automated detection of craters [6,7], which are regarded as significant landmarks on Mars and can also provide useful information about the geological processes. Other kinds of impact landforms are also considered as target objects in some researches, such as volcanic rootless cones [8], gully [9], and dark slope streak [10]. In these studies, involved object detection algorithms are categorized as highlight and shadow region matching [11], curve fitting [12], mathematical morphology [9,13], texture analysis [14], and machine learning [8,15–17]. While promising and showing good performances in their case studies, these approaches have some common flaws: due to the limitation of morphology-based detection models or low capacity of shallow machine learning models, little work has been done to develop generalized detectors for multiple geological landforms; most proposed detection algorithms rely on pre-defined hand- crafted features or geo-object-based features as their input, which require complicated feature selection

© 2021. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/

and extraction operations. Sometimes these pre-defined features might be unrepresentative, incomplete or redundant. Sliding window is widely used to extract region of interest in previous machine learning-based detectors, which is quite time-consuming and cannot meet the requirement of onboard processing system.

Generally, the challenges of developing object detection algorithms for Martian landforms can be summarized thus. First, in those remote sensing images captured by orbiters or spacecrafts, the same landform may present multiple scales within the extent of a single image scenes. Target objects in remote sensing images are relatively small compared with objects in nature images and the background environments are more complex. Second, the contour of a landform may possibly appear similar to another kind of landform. As shown in Fig. 1, both craters and cones show similar the circle-like shape and contain similar shadow on the edge. In addition, for some bar-shaped landforms, the ratio between width and length may vary dramatically. Third, as the remote sensing images are captured from the top view, target landforms are rotated arbitrarily around the vertical axis, which makes it inappropriate to use rectangle box with fixed orientation to locate certain landforms. Fourth, although an in- creasing number of remote sensing images captured by onboard cameras on orbiters have opened to public, available training images with expert annotations are significantly more limited and less accessible than popular data for many computer vision problems. With limited data and annotation, the capability of existing object detection algorithms based on machine learning would be constrained and potential over-fitting phenomenon might be difficult to avoid [18]. These four challenges make it very difficult to develop an object detection algorithm that can localize and identify multiple target landforms in the image scene, based on template, morphology, or hand-crafted features. Therefore, a new framework with ample model capacity needs to be introduced in this task, in order to design an efficient landform detection algorithm.

Recently, deep learning techniques have achieved dramatic progress in object detection and other kinds of computer vision applications [19–23]. With the help of powerful feature extraction ability from convolutional neural network (CNN) and the combi- nation of large-scale data sets and high-performance computing hardware GPUs, deep learning-based object detection models have obtained faster speed and higher accuracy than traditional methods in manifold application fields [24,25]. The deep CNN is able to automatically learn and discover important features from input images in model training process, instead of depending on pre-defined hand-crafted features in the conventional pattern recognition workflow. The number of object detection algorithms containing deep CNN have exploded in the last five years, from region proposal-based methods (e.g. R-CNN [26], fast R-CNN [27], and faster R-CNN [20]) to regression-based methods (e.g. Yolo [21] and SSD [22]). Although presenting good performance, these algorithms use a horizontal bounding box to locate a certain object, which is not suitable for landforms with arbitrary orientations in remote sensing images. Besides, model with deep architecture may demand a huge number of training data, which is unrealistic in this specific field. The transfer learning strategy has recently been proposed in deep learning techniques to solve this problem, by transfer the knowledge across different domains [28]. However, the gap between the remote sensing domain and the nature scene domain remains significant, so it may be not easy for a landform detection model to inherit some promising initializations from existing well-trained deep architectures.

To address these problems, in this study, we propose an end- to-end detection algorithm, named as Rotated-SSD (R-SSD), to detect two common landforms on Martian surface by using rotated bounding boxes with arbitrary orientations. The two landforms are defined as bar-shaped buttes and dark slope streak (DSS), which are two distinct landmarks and meaningful for motion estimation system and geological study. A butte is defined as an isolated hill with steep (often vertical) sides and can be understood as small mesa or plateau. A DSS is normally seen as narrow, dark, fan-shaped features and it normally appears on dust-covered slopes. Each DSS varies in the ratio of length to width and often appears very near to adjacent ones and thus cannot be easily detected by traditional sliding window-based methods. The objective of this study is to develop a single detector for these two landforms, so the difficulty of the task is further increased. Our detection algorithm develops from the Single Shot MultiBox Detector (SSD) model [22], which generates a set of default boxes with various ratios and scales from different convolutional layers to simultaneously predict the location and type of the target object. In the present algorithm, we add an angle parameter to each default box and in this way our algorithm can detect landforms in an end-to-end way using rotated bounding boxes, which is extremely fast. The default boxes are generated from multiple feature maps and the corresponding predicted results are combined at the final layer. This mechanism ensures the accuracy of landform detection. Overall, the main contributions of this study can be summarized as follows:

(1) An angle parameter is added to both the ground truth box and the default box and a new match strategy is designed for anchoring each default box to ground truth boxes in the model training process. With the match strategy, each default box is regarded as positive, negative, or neutral box and then its offset can be calculated via the pre-defined loss function. This operation plays a crucial role in training the entire deep neural network, but to our best knowledge, there is little work exploring the mechanism and effectiveness of match strategy.
(2) An autoencoder-based deep CNN is introduced for the model pre-training. This operation is essentially an unsupervised learning process and is conducted before the normal model training. By the pre-training operation, the base network of R-SSD is expected to get a promising initialization from the pre-trained network, which enhances the efficiency of the normal model training. To some degree, this unsupervised pre-training strategy is equivalent in function to the transfer learning, without introducing irrelevant information from the nature scene domain.
(3) All images used for model training and performance evaluation are collected from the High-Resolution Imaging Science Experiment (HiRISE) database. The camera of this project can take pictures of Martian surface with resolutions of 0.3 m/pixel, which makes it possible to simulate a closer look at impact landforms on Mars. The training and testing images are collected from different positions of HiRISE images with multi-scales, with the purpose of simulating the real scene of Mars captured

by the onboard cam- eras in landing procedure. This design ensures the reliability and practicality of evaluation results. Since the present detector does not explicitly elaborate the feature design and feature extraction, the deep learning-based framework can be easily generalized to other kinds of landforms or rotated objects.

The rest of this paper is organized as follows. Section 2 reviews the related works of this study. Section 3 describes the detail of our methodology, including the framework of R-SSD and details of the proposed match strategy and pretraining strategy. Section 4 presents and discusses the evaluation results. Section 5 concludes the paper.

2. Related works

2.1. Landform detection methods

Among existing geographic object-based image analysis and object detection researches, crater detection is frequently concerned, because of the useful information on geological processes provided by impact craters [11,29,30]. Other common types of landforms, such as volcanoes, rootless cones [8], dark slope streaks [10], or gullies [9], are also set as the target in some approaches. From the view of object detection algorithm, the morphology-based algorithms and machine learning-based algorithms are the most widely used. As for morphology-based algorithm, a single type of landforms is detected based on its unique mathematical morphology-based features, such as the linear features of gully or the circular or elliptical shape of crater. For example, Vamshi et al. [29] designed an object-based image analysis (OBIA) algorithm to detect impact craters on the Moon, in which the interested objects are extracted by multiresolution segmentation and their circularity and slope information are calculated for classification. Zhou et al. [30] designed a new crater detection algorithm by extracting higher change rate of slope of aspect values at crater rims. The noise of non-crater rims were filtered based on the neighborhood mean algorithm and reclassification method. As for the machine learning-based algorithms, most relevant approaches used the sliding window- based strategy to extract regions of interest (RoI), i.e., a square window with fixed or variable size slides across the entire input image to generate plenty of regions of interest. Then, features can be extracted from these regions as the input for machine learning models, by using pre-defined feature description methods. For instance, Burl et al. [3] introduced multiple machine learning models for crater detection, with training and testing examples generated by sliding windows. Limited to the fixed angle of each sliding window, these methods cannot be easily generated to landforms with arbitrary orientations.

Some researches attempted to use morphology-based strategies instead of sliding window to generate more flexible RoIs: Wang et al. [10] designed a novel DSS detection method by combining a new region extraction algorithm and machine learning techniques. Xin et al. [31] extracted impact sites candidates from full HiRISE images by dark area extraction and a series of morphological operations and then the AdaBoost classifier is trained with a cascade of features calculated from the candidates. How- ever, the morphology-based region extraction algorithm requires complex pre-processing operation and is restricted to the unique characteristic of a single type of landform. Hence, to design a general object detection algorithm for different landforms with arbitrary orientations, both the RoI extraction algorithm and the machine learning model need to be improved.

2.2. Deep convolutional neural network-based object detection methods

With the success of AlexNet [32] in ImageNet Large Scale Visual Recognition Challenge, deep CNN-based frameworks have achieved dramatic progress in the following years, in computer vision-related tasks [33,34]. Starting from R-CNN, deep CNN- based object detection methods exhibit high performance in detecting common objects in normal images, by adopting a two- stage pipeline: region proposal and object classification. This type of detection methods is named as region proposal-based net- works and consists of R-CNN, Fast R-CNN [27], Faster R-CNN [20], R-FCN (region-based fully-convolutional network) [35], etc. To further decrease the model inference time, regression-based net- works are proposed by using a single CNN-based network to generate bounding boxes and object probabilities simultaneously in one pass, including YOLO [21] and SSD [22]. Since it is extremely laborious to manually annotate object bounding boxes, image sets with object-level annotations are quite limited and valuable. Hence, transfer learning techniques are frequently used in order to train deep CNN with small-scale data sets, while applying these networks to solve different object detection tasks [36– 39]. However, all these networks are designed for objects in nature scene, which appear in the horizontal or vertical direction. Recently, some deep CNN-based networks are re-designed for generating bounding boxes with arbitrary orientations and they are applied for ship, airplane and vehicle detection in remote sensing images [40–42]. For example, Tang et al. [43] designed an end-to-end method to detect the vehicle's localization and orientation from aerial imagery data sets. Similarly, a rotated region proposal network (R2PN) is proposed to generate multi- orientated proposals with orientation angle information for ship detection from remote sensing images [42]. As far as we know, there is no well-designed framework for detecting landforms with arbitrary orientations.

3. Methodology

3.1. Overall network structure of R-SSD

The overall structure of the proposed R-SSD is illustrated in Fig. 2. This is an end-to-end detector: the input image goes through deep CNN to generate detection results (both object class score and location) in one pass. The entire deep framework can be divided into two parts: base network and extra feature layers, according to their different functions. The lower 7 layers are the base network, which is a modified version of VGG-16 network. Similar with VGG-16, the base network consists of 7 pairs of convolutional (*C*) layers and max-pooling (*M*) layers, but the fully connected layers are removed at the end of the base network. This modification can enhance the connectivity between the base network and extra feature layers whilst making the entire architecture more condensed. The base network plays a significant role in extracting useful features from input images and learning feature representations for the downstream extra feature layers. Based on some initial comparative tests, we have found that 7 *C* and *M* layers is a suitable setting for our task, which can drive the training and validation error to converge on a reasonable interval after the model training.

The extra feature layers can generate default boxes with different sizes (controlled by the size of convolutional layer), aspect ratios, and angles, which are used for searching and locating ground truth boxes. For a single extra feature layer, it is a grid firstly installed with a fixed kernel size of 3 * 3. Each kernel can generate k rectangular default boxes at center of itself. In each kernel, the size of default boxes is determined by two pre- defined hyper-parameters: aspect-ratio and scale. k is determined by both the number of aspect-ratios per layer and the multiple angles of each default box. All default boxes generated from different layers are assembled at the end of extra feature layers and then a matching process is executed to anchor the massive number of default boxes on ground truth boxes. In the matching process, if the overlap between a default box and a ground truth box is larger than a threshold, then the default box is identified as a positive one. All downstream operations, including obtaining offsets between default boxes and ground truth boxes, loss function and gradient calculation, and weights update of entire model, are based on matching results. Therefore, a well-designed match strategy for estimating the overlap between two bounding boxes with arbitrary orientations can significantly enhance the performance of model training and inference. More details of the proposed match strategy are described in the following section.

3.2. Match strategy

As shown in Fig. 2, we add an angle parameter to the rectangular bounding box with the purpose of making detection bounding boxes oriented. For both rotated default boxes and ground truth boxes, they are parametrized by five tuples (x_c , y_c , w, h, θ), where (x_c , y_c) is the center of the rotated bounding box, w, h is the length of the short side and the long side, respectively (See Fig. 3(a)). $\theta \in \{ 0, \pi/6, \pi/3, \pi/2, 2\pi/3, 5\pi/6, \pi \}$ is the orientation of the long side with respect to the y-axis. These multi-angle default boxes can cover almost shapes of landforms, with different sizes and aspect ratios. Given a default box D and a ground truth box GT, it is important for a match strategy to evaluate their distance and overlap, which is used to select positive and negative default boxes for calculating the loss function. The most commonly used criterion is the Intersection-over-Unior, which is defined as is the orientation of the long side with respect to the y-axis. These multi-angle default boxes can cover almost shapes of landforms, with different sizes and aspect ratios. Given a default box D and a ground truth box GT, it is important for a match strategy to evaluate their distance and overlap, which is used to select positive and negative default boxes for calculating the loss function. The most commonly used criterion is the Intersection-over-Unior, which is defined as

$$IoU\,(D, GT) = area(D \cap GT)/area(D \cup GT) \quad (1)$$

However, unlike the case for two vertical or horizontal boxes, the Boolean calculation for two rotated boxes is much more complex because the intersection of two rotated boxes can be any polygon with no more than eight sides [44], as shown in Fig. 3(b), (c), and (d). To solve this problem, an efficient two-stage match strategy is proposed in this study, which estimates the approximate IoU between two rotated boxes by concerning both the distance and the angle offset. Details of the match strategy is illustrated in Fig. 4. First, the angle of two rotated boxes D and GT are ignored and the non-rotated IoU is defined as:

$$NRIoU\,(D, GT) = area(D^{\wedge} \cap G^{\wedge}T^{\wedge})/area(D^{\wedge} \cup G^{\wedge}T^{\wedge}), \quad (2)$$

where where $\hat{D}$ and $\hat{G}\hat{T}$ keep the same shape with D and GT but rotated back to the vertical orientation. The criterion NRIoU efficiently estimates the overlap ratio between two rotated boxes because the overlap of two vertical bounding boxes is definitely a rectangle. If the NRIoU of D and GT is larger than a predefined threshold T1, then their angle difference is calculated. Only one D with minimum angle difference on that position will be matched to the corresponding GT if the θT is set to $\pi/12$. The entire matching process is actually a two-stage filtering process for picking up positive default boxes and the selection scope is narrowed down after the first stage. In this way, the computational cost is significantly reduced and the quality of matching results is still maintained. If the NRIoU of D and GT is smaller than T2, then the D is recognized as a negative default box (i.e., background). All remaining default boxes are recognized as neutral and irrelevant to the loss function calculation, because they are too close to a GT to be valid background boxes. In this study, T1 and T2 are set as 0.4 and 0.2, respectively, based on exhaustive initial tests.

3.3. Loss function and learning algorithm

In model training process, a loss function is calculated for weights update after every matching process. All positive and negative default boxes with associated ground truth boxes are involved for the calculation. Overall, the objective of the loss function is defined as a weighted sum of object confidence loss and localization loss:

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \quad (3)$$

where N is the number of matched (positive) default boxes; $x_{ij}^{p} = \{0, 1\}$ is an indicator variable of matching ith default box to jth ground truth box of class p. α is a weight term determining trade- off between two loss terms. Same as the original SSD model, the object confidence loss L_{conf} is the softmax loss over multiple classes confidences c:

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0), \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \quad (4)$$

The localization loss L_{loc} is a Smooth L1 loss between the predicted box (l) and the ground truth box (g). Here we add the angle difference of D and associated GT to the localization loss of original SSD and regress to offsets for the center (cx; cy), width (w), height (h) and angle (θ) of the default bounding box (d):

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h, \theta\}} x_{ij}^{k} \, \mathrm{smooth}_{L1}(l_i^m - \hat{g}_j^m),$$
$$smooth_{L1}(x) = \begin{cases} 0.5x^2 \ if \ |x| < 1 \\ |x| - 0.5 \ otherwise \end{cases}, \quad (5)$$
$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w, \quad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h,$$
$$\hat{g}_j^w = \log(g_j^w / d_i^w), \quad \hat{g}_j^h = \log(g_j^h / d_i^h), \quad \hat{g}_j^\theta = g_j^\theta - d_i^\theta.$$

Due to the huge number of default boxes generated from multiple extra feature layers, most default boxes are negative ones. Using all these negative boxes with limited positive ones will lead to a significant imbalance between positive and negative training examples. To overcome this problem, a hard-negative mining strategy [22] is introduced, which fixes the ratio between positive and negative default boxes to 1:3 by only using negatives with high confidence loss. This screening process can make the weight optimization more efficient and stable.

Unlike the original SSD, we introduce a more robust optimizer Adam as the model learning algorithm. Adam is a variant of the stochastic gradient descent algorithm and its details are described in the reference [45]. In Adam optimizer, the exponential moving averages of gradients and squared gradients are updated in every epoch, while their exponential decay rates are controlled based on two hyper-parameters. This optimizer can be understood as a combination on advantages of two popular algorithms: Ada- Grad [46] and RMSProp [47] with some important improvements. One is that the momentum is determined as an estimate of the first-order moment of the gradient. Another one is that the Adam adds bias corrections to the estimates of both the first-order moments (i.e., the momentum term) and the second-order moments, which help to counteract their initial biases towards zero. Generally, the advantages of Adam optimizer can be summarized as invariant magnitudes of weight update with respect to rescaling of the gradient and bounded learning rate by hyperparameters.

3.4. Autoencoder-based pretraining strategy

Although the convolutional deep architectures have been successfully applied to various areas, it is very difficult to train them with small scale data sets due to some optimization challenges. For example, the weight in lower layers is more difficult to get converge compared with upper layers, because of the gradient vanish phenomenon [48]. Similar to most of non-convex optimization problems, deep CNN-based architectures contain enormous local minima and it is very likely that model parameters can become stuck in the same minima during the training process. Therefore, it is meaningful to train deep architectures from promising initializations instead of scratch (random initialization). To solve this problem, most relevant studies on object detection have applied a transfer learning strategy, in which a base network is trained on a base data set (normally a large image set with various classes of objects). Then learned features (i.e., model parameters) in the base network is set as the initialization for the target network, which will be trained on a target data set and task. This strategy will tend to work in cases where features are suitable to both base and target tasks. However, there exists an obvious gap between Martian surface images taken by orbiters and popular object detection data sets (e.g., COCO), which are taken from everyday scenes. In this study, we design an autoencoder-based pretraining strategy to obtain a promising initialization, which is actually an unsupervised learning process and can learn intrinsic features from unlabeled HiRISE images.

The main idea of the strategy is to train a deep autoencoder which can reconstruct the input images. The structure of the proposed autoencoder network is shown in Fig. 5, which is a fully convolutional deep neural network. The encoder part is set as the same as the base network of R-SSD, which consists of 7 convolutional layers with two max-pooling layers. The decoder

consists of 7 deconvolutional layers with two up-sampling layers, which are the reverse of the encoder with no tied weights. In de- convolutional layers, the learned filters serve as the foundation to reconstruct the information of the input. The up-sampling layers act as the reverse operation of max-pooling and it can reconstruct the original convolutional kernel. In this way, the autoencoder can output a tuple with the same size as the model input. The mechanism of the autoencoder is under a hypothesis that the network can spontaneously abstract essential information from input images, by learning relevant features of input images with low reconstruction error. Training the autoencoder does not need annotation information of the input data. The objective is to minimize the re-construction error between input and associated output, by using the back-propagation algorithm to adjust the parameters of both encoder and decoder. After the autoencoder is pre-trained, the parameters in the encoder part is saved and used as initialization for training the R-SSD.

4. Experiments and results

4.1. Image data preparation

All training and testing images are collected from the HiRISE image library. HiRISE is the most powerful camera ever sent to Mars, one of six instruments onboard the Mars Reconnaissance Orbiter. The camera is designed to capture surface features of Mars in greater detail than has previously been done from orbit. The high-resolution capability of HiRISE allows better studies of Martian landforms and stimulate the development of object detection algorithms for Martian landforms localization and identification. As the target objects are set as buttes and DSSs in this study, we selected 13 map-projected HiRISE images with high density of buttes or DSSs (see Fig. 6) as raw materials for generating training and testing data : ESP_011966_1700, ESP_013544_1995, ESP_017411_1715, ESP_011289_1950, ESP_012383_1905, ESP_014394_2045, ESP_015937_1760, ESP_028642_1800, ESP_036851_1995, ESP_040386_1915, ESP_043523_2040, ESP_038343_1785, PSP_003570_1915. For each selected HiRISE image, it is firstly downsized to a series of images with low resolutions (width equal to 800, 1200, 1600, 1800, 2000, 2400, 3000, and 3200 pixels). Then, all training and testing images with fixed size (300*400 pixels) are collected from these downsized images. Some images in training and testing sets are collected from the same raw material but have different scales and the corresponding image group simulates the real scene captured by the onboard camera at different altitude during the landing procedure (see Fig. 7). Moreover, the robustness and the universal property of R-SSD are enhanced as the model is trained on varying object sizes. Table 1 shows the number of images and target objects in each data set. The training set and testing set I are collected from first eleven original HiRISE images and the testing set II is collected from the remaining two original HiRISE images. All images in the training set is annotated manually by the authors, by using an efficient tool named as roLabelImg (downloadable from https://github.com/cgvict/roLabelImg). The roLabelImg is a graphical image annotation tool which can label arbitrarily rotated objects with rotated rectangular bounding boxes. Each box is recorded by using five parameters: the x and y coordinates of its center point, the width, the length, and the angle deviation from the vertical direction.

4.2. R-SSD implementation and evaluation

The implementation of R-SSD is based on a Keras implementation of SSD architecture [49], by adding the proposed strategies and modules. The number of total parameters of R-SSD is 981,766 and the number of floating-point operations per second is 23502k. All involved comparative experiments are written in Python 3.6 and run on a workstation equipped with 3.7 GHz Intel ® Core ™ i7-8700 K CPU, 64-GB RAM. dual NVIDIA® GTX 1080Ti GPUs. To evaluate the effectiveness of proposed individual modules and overall efficacy of R-SSD, the experimental evaluations of our deep architecture-based landform detection system are conducted in three parts. In the first part, the effectiveness of proposed match strategy is investigated, by comparing it with other 4 types of state-of-the-art match strategies for anchoring default boxes to ground truth boxes. Second, the proposed autoencoder-based pretraining strategy is evaluated, by comparing it with non-pretrained R-SSD, R-SSD pretrained on common image sets and transfer learning. In this part, the training error in the model fine-tuning stage is monitored epoch by epoch to investigate the effect of pretraining strategy on the normal model training. In the third part, the overall performance of R-SSD on two types of landforms is qualitatively illustrated. For comparative experiments in first two part, all involved methods are evaluated on both testing set I and II, and an IoU- based criterion is applied to judge if a predicted box is a true positive detection: by calculating the IoU between a given predicted box and any ground truth box, each predicted box is recognized as either true-positive or false-positive. As positive and negative detections are defined, the average precision (AP) is introduced as the quantitative metric for model performance, like the model evaluation in most object detection tasks. AP measures the area under the precision–recall curve and in this study, it is approximated by a finite sum over positions in the ranked sequence of detections:

$$AP = \int_0^1 p(r)dr \approx \sum_n (R_n - R_{n-1})P_n \qquad (6)$$

where Rn and Pn are the recall and precision at the nth threshold.

4.3. Evaluation of proposed match strategy

To evaluate the efficacy of the proposed two-stage match strategy, four state-of-the-art match strategies are considered here, which are abstracted from the three most relevant approaches. The Match Strategy 1 (MS1) is a center point-based method, based on Tang et al.'s approach [43]. The criterion can be summarized as: for any ground truth box, if the center of a default box is inside a ground truth box and the ratios between the default box and the ground truth satisfy Eq. (7), then the default box is recognized as positive. In Eq. (7), w_d and h_d denote the width and height of the default box and w_{gt} and h_{gt} denote those of the ground truth boxes.

$$\max\left(\frac{w_d}{w_{gt}}, \frac{w_{gt}}{w_d}, \frac{h_d}{h_{gt}}, \frac{h_{gt}}{h_d}\right) \leq 1.5 \tag{7}$$

The Match Strategy 2 (MS2) is a variant version of MS1, in which the size related criterion is replaced with the angle difference of two boxes, i.e., the Eq. (7) is replaced with:

$$|\theta_d - \theta_{gt}| \leq \pi/12 \tag{8}$$

where θ_d and θ_{gt} denote the angle of the default box and the ground truth box, respectively. The Match Strategy 3 (MS3) is abstracted from Liu et al.'s work [44] and it is described as: a default box D is assigned to a ground truth box GT if $ArIoU(D, GT)$ is larger than a pre-defined threshold. The $ArIoU$ is defined as Eq. (9), which merges the distance difference with angle difference.

box D is assigned to a ground truth box GT if $ArIoU(D, GT)$ is larger than a pre-defined threshold. The $ArIoU$ is defined as Eq. (9), which merges the distance difference with angle difference.

$$ArIoU(D, GT) = \frac{area(\hat{D} \cap \hat{GT})}{area(\hat{D} \cup \hat{GT})} \cos(\theta_D - \theta_{GT}) \tag{9}$$

The Match Strategy 4 (MS4) is derived from Xia et al.'s work [42] and based on 4 vertexes of a rotated rectangular box. Let (x_D^i, y_D^i) and (x_{GT}^i, y_{GT}^i) denote the coordinates of the ith vertexes of a default box and a ground truth box. If their mean deviation is less than a threshold (see Eq. (10)), then the default box is considered as positive and matched to the associated ground truth box.

$$mean(|x_D^1 - x_{GT}^1|, |x_D^2 - x_{GT}^2|, |x_D^3 - x_{GT}^3|, |x_D^4 - x_{GT}^4|) < T \tag{10}$$

All these four match strategies and the proposed match strategy (MS) are embedded in the same deep architecture (i.e., R-SSD) and trained with the same data set under the same configuration: the number of epochs is set to 20 with 300 steps per epoch and the batch size is set as 20. Other significant hyper-parameters (e.g., scales and aspect ratios of default boxes, T1, T2, and T) are optimized via the grid search algorithm. Comparison results with the model training time and model inference performance on two testing sets are shown in Table 2. Specially, the proposed MS takes less training time than MS1, MS2, and MS4, which demonstrates the high efficiency of the two-stage filtering mechanism and NRIoU-based location deviation estimation method. MS3 achieves comparable training time whereas worse performance than MS1 for the detection of DSS on the testing set II. This may indicate that converting two kinds of objectives into one criterion may have side-effects on model accuracy. In addition, can be observed that MS outperforms MS2, MS3, MS4 in most assessment cases, which shows the high accuracy of MS. MS1 obtains comparable AP in testing set II but performs bad in testing set I and hence further suggests the effectiveness of the proposed MS.

4.4. Evaluation of the pretraining strategy

The non-pretrained R-SSD is firstly compared with pretrained R-SSD using similar HiRISE images. The pretraining data set is collected from the training set and similar HiRISE images containing high density of buttes or DSSs. With regard to the proposed autoencoder-based pretraining strategy, all these images are un- labeled. To further test the impact of number of pretraining images on model performance, we pretrain the R-SSD with 200, 400, and 600 rescaled HiRISE images, respectively. In the pretrain- ing stage, the batch size is set as 25 and the number of epochs is set as 200. The loss function values of non-pretrained R-SSD and pretrained R-SSD during the model fine-tuning process are monitored and plotted in Fig. 8.

In Fig. 8, pretrain_200, pretrain_400, and pretrain_600 de- note that the corresponding model is pretrained with 200, 400, and 600 rescaled HiRISE images, respectively. The loss function value of three pretrained models is lower than that of the non- pretrained model in the first epoch, which means that this un- supervised pretraining operation can still help model to get into a promising state at the beginning of finetuning stage. From the loss function values in each epoch, we cannot find any significant difference among pretrain_200, pretrain_400, and pretrain_600. Although there is an abnormal point in the model fine-tuning process of pretrain_600 (12th epoch), the loss function value of pretrain_600 is still lower than that of non-pretrain at the final epoch (0.319 vs 0.400). Detection results of these models on two testing sets are listed in Table 3. Based on the result, a significant improvement of AP is achieved by pretrained models compared with the non-pretrain model. Although the difference of AP is not significant with respect to testing set I, all pretrain_200, pretrain_400, and pretrain_600 obtain higher AP on both buttes and DSSs detections

with respect to testing set II, compared with the non-pretrain. Hence, the autoencoder-based pretraining strategy has a remarkably positive effect on model inference and the quality of detection results. Pretrain_400 outperforms pretrain_200 on detection of buttes and DSS in both two testing sets. However, pretrain_600 obtains very low AP on detection of DSS in testing set II, though it performs well on detection of buttes. The unbalanced results indicate the low stability and universal property of pretrain_600. Therefore, the scale of pre- training set should be controlled in a reasonable range and in this study pretrain_400 is selected as final model and used for further experiments.

To give insight on the intermediate results learned by the autoencoder-based pretraining process, we visualize the filters learned by the autoencoder of the 3rd convolutional layer with and without pretraining. Fig. 9 shows nine visualized filters of one example in the training set, learned by the autoencoder with pretraining; Fig. 10 shows the same nine visualized filters in the autoencoder without pretraining. By comparing these two figures, it is revealed that the autoencoder with pretrained weights to process the input data can learn edges and skeletons of most buttes whereas the autoencoder with random-generated weights just blurs the input images. The visualization of intermediate results corroborates the effectiveness of the pretraining strategy.

To evaluate the influence of image property on the autoencoder-based pretraining strategy, we introduce two bench- mark large-scale image sets, COCO and KITTI, for model pretraining and compare their performance with pretrain_400. COCO [50] is a large-scale object detection, segmentation, and captioning data set containing 80 complex everyday scenes of common objects. KITTI [51] consists of real-world traffic situations captured by cameras and laser scanners on autonomous vehicles driving through a city. These two data set are widely used for object detection model evaluation. The loss function values of R-SSD pretrained from pretrain_400, pretrain_COCO, and pretrain_KITTI are plotted in Fig. 11 and detection results of these models are shown in Table 4.

From Fig. 11, it can be observed that the loss function values of both pretrain_COCO and pretrain_KITTI is distinctly higher than those of pretrain_400 during the first five epochs. This phenomenon means that the autoencoder pretrained on COCO or KITTI data set can hardly learn anything useful for our task, so it cannot provide a promising initialization for R-SSD fine-tuning. From Table 4, pretrain_COCO and pretrain_KITTI still perform worse than pretrain_400, especially in testing set II. Additionally, pretrain_KITTI only results in 0.082 AP on detection of DSS in testing set II. Therefore, it is summarized that autoencoders pre-trained on COCO and KITTI bring negative effect on the model performance, probably because the underlying domain gap be-tween HiRISE images and daily scenes may be too large to benefit the detection performance.

To compare the autoencoder-based pretraining with the widely-used transfer learning scheme, we employ model weights of the original SSD330 trained from three large-scale data sets, which are the Pascal VOC [52] (PASCAL Visual Object Classes), the COCO, and the ILSVRC [53] (ImageNet Large Scale Visual Recognition Challenge). The transfer learning is implemented as follows: the base network of R-SSD is firstly replaced with one of these pretrained base network, which is the reduced VGG-16 network with model weights converged to a certain data set. Then, the model fine-tuning is conducted by training the modified R-SSD with our own data set (the training set). Based on the data set used in transfer learnings, we denote these three-transfer learning schemes as TL_Pascal_VOC, TL_COCO, and TL_ILSVRC. The associated efficiency of model training and detection results are shown in Fig. 12 and Table 5. From Fig. 12, the loss function value of TL_Pascal_VOC, TL_COCO, and TL_ILSVRC is distinctly higher than that of pretrain_400 during the entire model fine-tuning stage, which indicates that all these transfer learning-based R- SSDs cannot reach convergence in the model fine-tuning process. These non-convergent models lead to extremely poor detection results as shown in Table 5. Therefore, it is risky to apply transfer learning scheme if the domain gap is too large.

4.5. Evaluation of the overall performance

To evaluate the overall performance of the proposed R-SSD, we employed two classical hand-crafted feature-based object de- tection algorithms and three types of well-known deep learning- based object detection framework. As for two hand-crafted feature- based method, one is morphology-based, in which each input image is firstly converted to grayscale image with binary matrix. Then, the rectangular structure element is extracted from the ob- tained binary map, followed by morphological erosion operation. After that, we conduct contour extraction operation to find the minimum outer of each rectangle area. Finally, all these rectangle boxes are collected and filtered with a pre-defined threshold on minimum area, maximum width and maximum height to generate the final detection results. We set different thresholds for two different objects, and each threshold is finetuned on the training set. Another method is color-based, in which each input image is firstly converted to HSV color space and the color-based segmentation operation is conducted on the current color space. Then, the segmentation result is filtered by erosion and dilation operation. Per-element bit-wise conjunction of filtered result and the corresponding original image is calculated to obtained the possible target of interest. Finally, similar with the first method, contours of rectangle areas are obtained by finding the minimum outer quadrilateral for each contour and the same filtering operation is conducted to get the final detection results. These two hand-crafted feature-based methods are implemented with OpenCV library and the performance on testing sets are listed in Table 6. As the two models work in unsupervised way and cannot output confidence interval for each predictive rectan-gle box, we calculate the accuracy of detection results on two testing sets as the evaluation metric. From Table 6, we notice that these two hand-crafted feature-based detection algorithms obtain extremely low accuracies, especially on detection of DSS in both two testing sets. The results further validate the predictive power of R-SSD and the difficulty of detecting butte and DSS from remote sensing images, as both targets varies in the ratio of length to width and DSSs often appear very near to adjacent ones and thus cannot be easily detected by traditional hand-crafted feature-based methods.

As for deep learning-based object detection framework, we employed two state-of-the-art end to end frameworks, namely YOLOv2 and YOLOv3, and a two-stage detector, namely Faster R-CNN as benchmark methods. To remove the inference of pre-training to model performance, these three models are trained from scratch and the testing results are compared with R-SSD without pretraining operation. To reduce the inference brought by model complexity, the base networks of three benchmark methods are adjusted to make their number of parameters sim- ilar to the proposed R-SSD. The number of multiply-accumulate operations and model parameters of these models are listed in Table 7. Comparative results shown in Table 8 indicate that both end-to-end and two stage detectors obtain lower AP than R-SSD, especially on detection of DSSs in two testing sets, which further validates the effectiveness of proposed match strategy for two bounding boxes with arbitrary orientations and the effectiveness of R-SSD on detecting two specific Martian landforms.

4.6. Qualitative analysis

Fig. 13 demonstrates some typical detection results achieved by the pretrained R-SSD and the input images are sampled from the testing set I and II. The ground-truth buttes and DSSs are an- notated by green rotated boxes and predicted results are denoted by light blue boxes (for buttes) and red boxes (for DSSs), with associated object name and confidence score. In Fig. 13(a) and (b), most buttes and DSSs are identified with relatively high fitting degree on both location and orientation, though two rugged areas are falsely detected as DSSs with low confidence values.

In addition, the pretrained R-SSD is capable of discriminating between buttes and DSSs appearing in complicated background. In, Fig. 13(c) to (f), although the accuracy of detection results is not as good as that in (a) and (b), most distinct objects are well identified with high confidence value. Especially in Fig. 13(e) and (f), even if some DSSs are densely located, the pretrained R-SSD can still correctly predict their locations and orientations in most cases. This suggests that the pretrained R-SSD maintains the high universal property to some degree, on detecting target objects in new backgrounds.

Several difficult cases that the pretrained R-SSD yields low-quality detections are illustrated in Fig. 14. From Fig. 14(a), it can be observed that distinct buttes cannot be identified by the proposed framework if they are surrounded by similar patterns or located in the environment full of hills. From Fig. 14(b) and (c), some DSSs cannot be detected in case that the surrounding environment is too dark or the DSS is faded. These poor performances indicated that the robustness of our system against challenging environments need to be further improved. In this paper, we mainly focus on the match strategy design and pretraining strategy design for training deep architectures with small-scale data. We leave other model performance improvement-related issues for future studies.

5. Conclusion

Distinctive of existing morphology-based or traditional ma- chine learning model-based Martian landform detection algorithms, the proposed deep neural network, R-SSD, aims to solve three major challenges: autonomous feature extraction and feature learning, objects with arbitrary orientations, and limited training data, for the automatic detection of two types of land-forms on Mars. To efficiently match arbitrarily oriented ground truth boxes with rotated default boxes, a two-stage match strategy is developed and executed in every epoch of model training. In addition, an autoencoder-based unsupervised pretraining operation is conducted before the normal model training and the pretrained encoder part is embedded in the base network of R-SSD to serve as a promising initialization. Our composite deep learning-based system has been evaluated with extensive comparative experiments and results demonstrate that both the match strategy and the pretraining strategy are helpful to improve the efficiency, performance and universal property of R- SSD. Instead of implementing a series of routine operations in traditional object detection algorithms (e.g., feature selection, region of interest extraction, classification,.. .), the R-SSD works in an end-to-end way, attributed to the proposed strategies and high-flexibility of CNN-based feature extractor.

Overall, the main contribution of this paper can be summarized as two folds: practically, the developed R-SSD may help to improve the onboard motion estimation system, by locating and identifying impact landforms more efficiently and accurately. Theoretically, this study would shed a light on the potential ability of composite deep architectures on the processing of objects with arbitrary orientations. Moreover, it may bring new ideas to the future studies on detecting other kinds of significant landforms, or even other common objects with arbitrary orientations.

CRediT authorship contribution statement

Shancheng Jiang: Conceptualization, Methodology, Software, Validation, Writing – original draft. **Fan Wu:** Data curation, Visualization, Writing – review & editing. **K.L. Yung:** Formal analysis, Investigation, Data curation, Writing – review & editing, Project administration, Funding acquisition, Resources. **Yingqiao Yang:** Writing – review & editing. **W.H. Ip:** Validation, Project administration, Funding acquisition. **Ming Gao:** Software, Data curation. **James Abbott Foster:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the project grant ZG3K Chang'e phase 3 sample return instruments, in part by the Na- tional Nature Science and Foundation of China Grand No. 71801031, in part by the Guangdong Basic and Applied Basic Research Foundation project, China, No. 2019A1515011962 and 2020A1515110431, in part by the National Nature Science and Foundation of China Grand No. 71772033, in part by the Natural Science Foundation of Liaoning Province, China (Joint Funds for Key Scientific Innovation Bases, 2020-KF-11-11), and in part by the Scientific Research Project of the Education Department of Liaoning Province, China (LN2019Q14).

References

[1] G. Chen, et al., Geographic object-based image analysis (GEOBIA): emerg- ing trends and future opportunities, GISci. Remote Sens. 55 (2) (2018) 159–182.

[2] L. Zhang, L. Zhang, B. Du, Deep learning for remote sensing data: A technical tutorial on the state of the art, IEEE Geosci. Remote Sens. Mag. 4 (2) (2016) 22–40.

[3] M.C. Burl, P.G. Wetzler, Onboard object recognition for planetary exploration, Mach. Learn. 84 (3) (2011) 341.

[4] R. Qin, W. Fang, A hierarchical building detection method for very high resolution remotely sensed images combined with DSM using graph cut optimization, Photogramm. Eng. Remote Sens. 80 (9) (2014) 873–883.

[5] D. Carrera, et al., Detection of sand dunes on mars using a regular vine-based classification approach, Knowl.-Based Syst. 163 (2019) 858–874.

[6] L. Bandeira, W. Ding, T.F. Stepinski, Detection of sub-kilometer craters in high resolution planetary images using shape and texture features, Adv. Space Res. 49 (1) (2012) 64–74.

[7] E. Emami, et al., Automatic crater detection using convex grouping and convolutional neural networks, in: Advances in Visual Computing, Springer International Publishing, Cham, 2015.

[8] L.F. Palafox, et al., Automated detection of geological landforms on mars using convolutional neural networks, Comput. Geosci. 101 (Suppl. C) (2017) 48–56.

[9] W. Li, et al., Automated detection of martian gullies from HiRISE imagery, Photogramm. Eng. Remote Sens. 81 (12) (2015) 913–920.

[10] Y. Wang, et al., Automatic detection of martian dark slope streaks by machine learning using HiRISE images, ISPRS J. Photogramm. Remote Sens. 129 (Suppl. C) (2017) 12–20.

[11] E.R. Urbach, T.F. Stepinski, Automatic detection of sub-km craters in high resolution planetary images, Planet. Space Sci. 57 (7) (2009) 880–887.

[12] G. Salamunićcar, et al., Hybrid method for crater detection based on topography reconstruction from optical images and the new LU78287GT catalogue of lunar impact craters, Adv. Space Res. 53 (12) (2014) 1783–1797.

[13] M. Chen, et al., Lunar crater detection based on terrain analysis and mathematical morphology methods using digital elevation models, IEEE Trans. Geosci. Remote Sens. 56 (7) (2018) 3681–3692.

[14] T. Barata, et al., Automatic recognition of impact craters on the surface of mars, in: Image Analysis and Recognition, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[15] S. Jin, T. Zhang, Automatic detection of impact craters on mars using a modified adaboosting method, Planet. Space Sci. 99 (2014) 112–117.

[16] T.F. Stepinski, D. Wei, R. Vilalta, Detecting impact craters in planetary images using machine learning, in: Intelligent Data Analysis for Real-Life Applications: Theory and Practice, IGI Global, Hershey, PA, USA, 2012, pp. 146–159.

[17] T. Vinogradova, M. Burl, E. Mjolsness, Training of a crater detection algo- rithm for Mars crater imagery, in: Proceedings, IEEE Aerospace Conference, 2002.

[18] H. Chen, et al., Ultrasound standard plane detection using a composite neural network framework, IEEE Trans. Cybern. 47 (6) (2017) 1576–1586.

[19] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[20] S. Ren, et al., Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015.

[21] J. Redmon, et al., You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[22] W. Liu, et al., SSD: Single Shot MultiBox Detector, in: Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016.

[23] Y. Wu, Z. Yi, Automated detection of kidney abnormalities using multi- feature fusion convolutional neural networks, Knowl.-Based Syst. 200 (2020) 13.

[24] G.B. Li, Y.Z. Yu, Contrast-oriented deep neural networks for salient object detection, IEEE Trans. Neural Netw. Learn. Syst. 29 (12) (2018) 6038–6051.

[25] Y.X. Tang, et al., Visual and semantic knowledge transfer for large scale semi-supervised object detection, IEEE Trans. Pattern Anal. Mach. Intell. 40 (12) (2018) 3045–3058.

[26] R. Girshick, et al., Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[27] R. Girshick, Fast R-CNN, in: 2015 IEEE International Conference on Computer Vision, ICCV, 2015.

[28] S. Hoo-Chang, et al., Deep convolutional neural networks for computer- aided detection: CNN architectures, dataset characteristics and transfer learning, IEEE Trans. Med. Imaging 35 (5) (2016) 1285.

[29] G.T. Vamshi, T.R. Martha, K. Vinod Kumar, An object-based classification method for automatic detection of lunar impact craters from topographic data, Adv. Space Res. 57 (9) (2016) 1978–1988.

[30] Y. Zhou, et al., Automatic detection of lunar craters based on DEM data with the terrain analysis method, Planet. Space Sci. 160 (2018) 1–11.

[31] X. Xin, et al., Automated detection of new impact sites on Martian surface from HiRISE images, Adv. Space Res. 60 (7) (2017) 1557–1569.

[32] A. Krizhevsky, I. Sutskever, G.E. Hinton, Magenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. (2012).

[33] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional net- works, in: Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014.

[34] C. Szegedy, et al., Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[35] J. Dai, et al., R-FCN: Object detection via region-based fully convolutional networks, 2016, pp. 379–387.

[36] Y. Tang, et al., Visual and semantic knowledge transfer for large scale semi- supervised object detection, IEEE Trans. Pattern Anal. Mach. Intell. 40 (12) (2018) 3045–3058.

[37] Á. Arcos-García, J.A. Álvarez-García, L.M. Soria-Morillo, Evaluation of Deep Neural Networks for traffic sign detection systems, Neurocomputing 316 (2018) 332–344.

[38] Y. Xu, et al., End-to-end airport detection in remote sensing images com- bining cascade region proposal networks and multi-threshold detection networks, Remote Sens. 10 (10) (2018) 1516.

[39] W. Liu, S. Chen, L. Wei, Improving street object detection using transfer learning: From generic model to specific model, J. Adv. Comput. Intell. Intell. Inform. 22 (6) (2018) 869–874.

[40] J. Wang, C. Lu, W. Jiang, Simultaneous ship detection and orientation estimation in SAR images based on attention module and angle regression, Sensors 18 (9) (2018) 2851.

[41] S. Li, et al., Multiscale rotated bounding box-based deep learning method for detecting ship targets in remote sensing images, Sensors 18 (8) (2018) 2702.

[42] Z. Zhang, et al., Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks, IEEE Geosci. Remote Sens. Lett. (99) (2018) 1–5.

[43] T. Tang, et al., Arbitrary-oriented vehicle detection in aerial imagery with single convolutional neural networks, Remote Sens. 9 (11) (2017) 1170.

[44] L. Liu, Z. Pan, B. Lei, Learning a rotation invariant detector with rotatable bounding box, 2017, arXiv preprint arXiv:1711.09405.

[45] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[46] J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, J. Mach. Learn. Res. 12 (Jul) (2011) 2121–2159.

[47] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA: Neural Netw. Mach. Learn. 4 (2) (2012) 26–31.

[48] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[49] P. Ferrari, A keras port of single shot MultiBox detector, 2018, [cited 2020 Aug 30th]; Available from: https://github.com/pierluigiferrari/ssd_keras.

[50] T.-Y. Lin, et al., Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014.

[51] A. Geiger, et al., Vision meets robotics: The KITTI dataset, Int. J. Robot. Res. 32 (11) (2013) 1231–1237.

[52] M. Everingham, et al., The pascal visual object classes (voc) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.

O. Russakovsky, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.

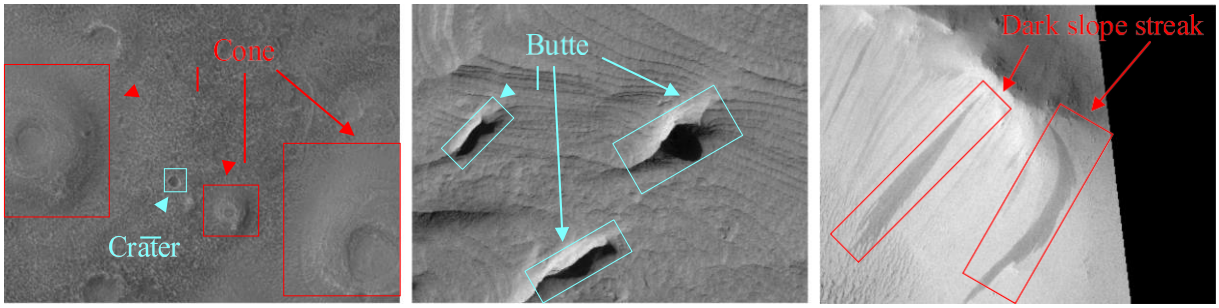


Fig. 1. Illustration of some common landforms on Mars.

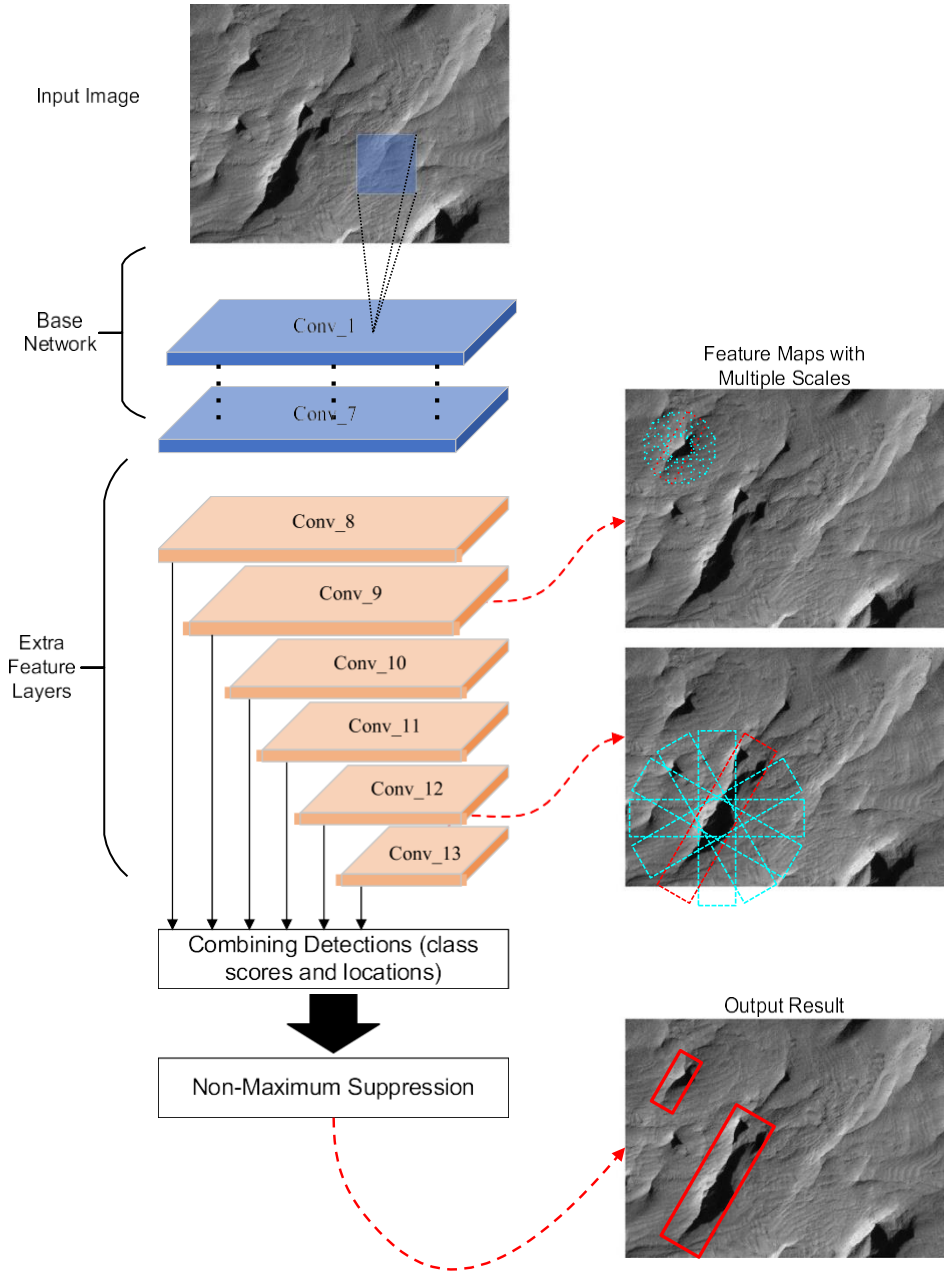


Fig. 2. Network structure of R-SSD

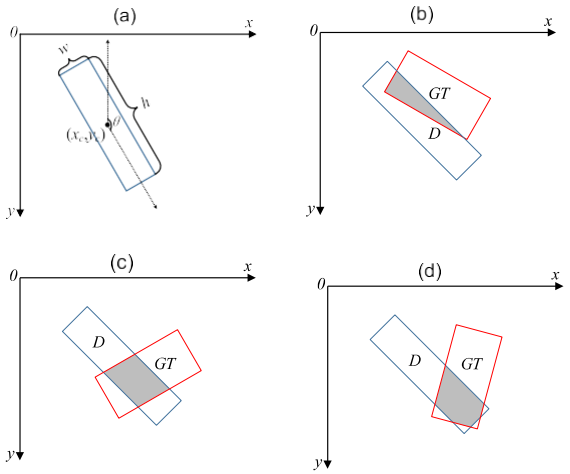


Fig. 3. Cases of the intersection of two rotated boxes.

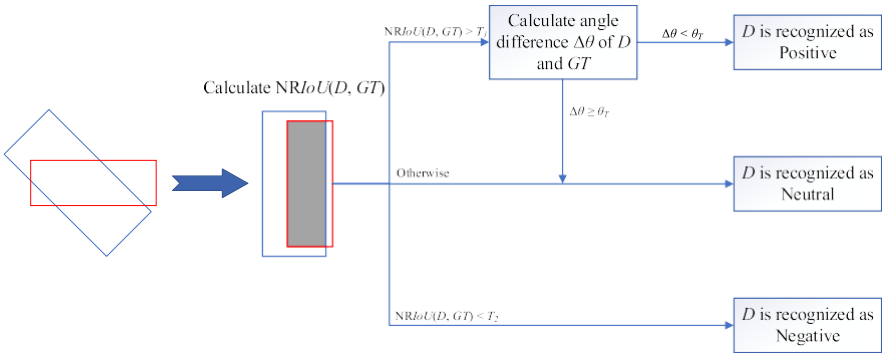


Fig. 4. Illustration of the proposed two-stage match strategy.

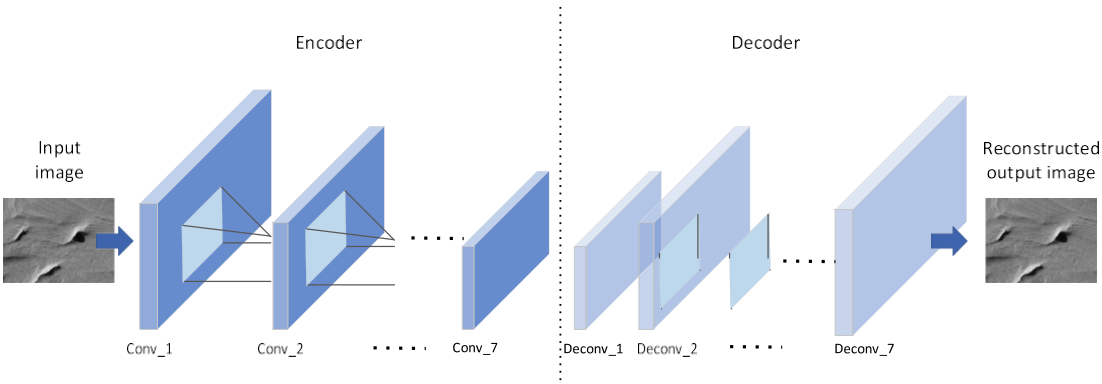
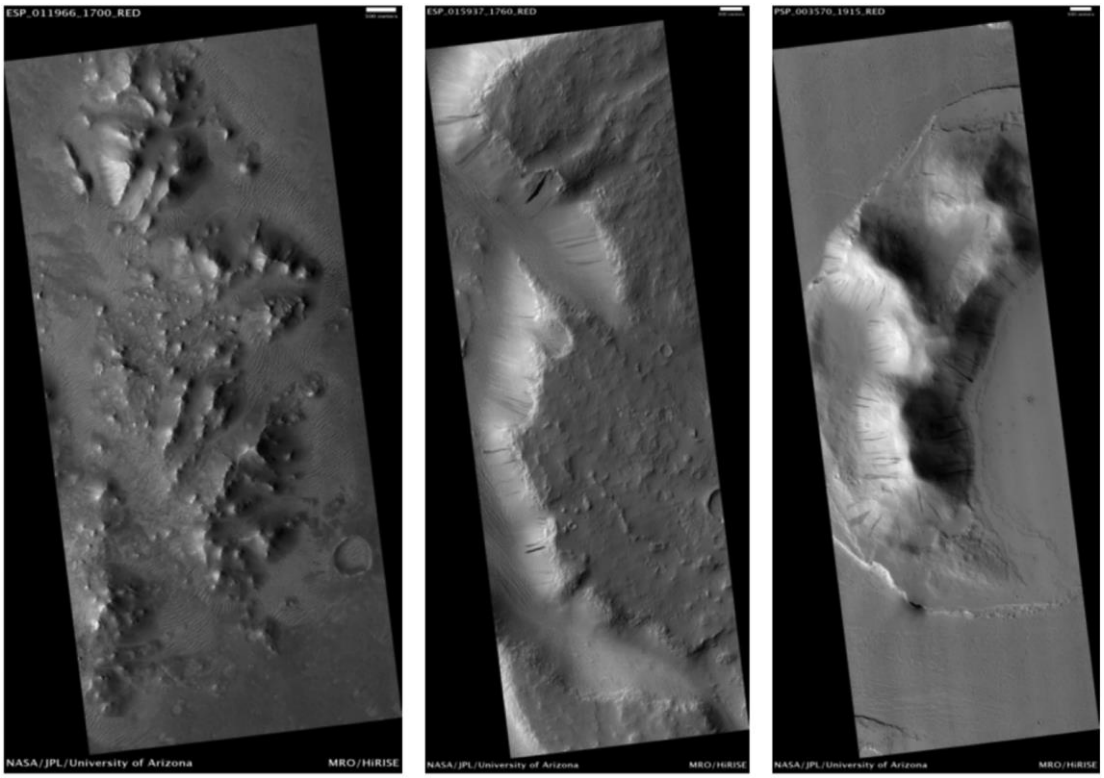


Fig. 5. Architecture of the deep autoencoder used in this work.



ID: ESP_011966_1700
Acquisition date: 13 February 2009
Latitude/Longitude: -9.733°/268.624°
Size: 26512*48622 (pixel)
Scale: 25 cm/pixel

ID: ESP_015937_1760
Acquisition date: 20 December 2009
Latitude/Longitude: -4.169°/216.673°
Size: 29013*66412 (pixel)
Scale: 25 cm/pixel

ID: PSP_003570_1915
Acquisition date: 01 May 2007
Latitude/Longitude: 11.346°/180.996°
Size: 30789*69528 (pixel)
Scale: 25 cm/pixel

Fig. 6. Illustration of three HiRISE images used in this study, all are downloadable from the HiRISE website https://www.uahirise.org.

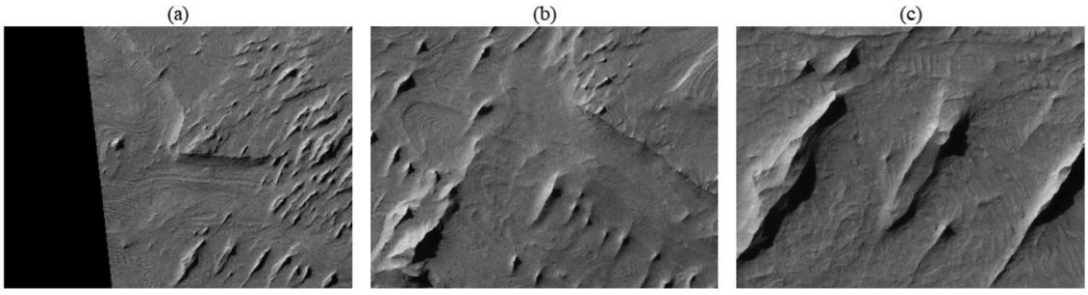


Fig. 7. Three examples in the training image set. The examples are collected from the same raw material but have different scales. The latter one can be regarded as an enlargement of the former image of a given field.

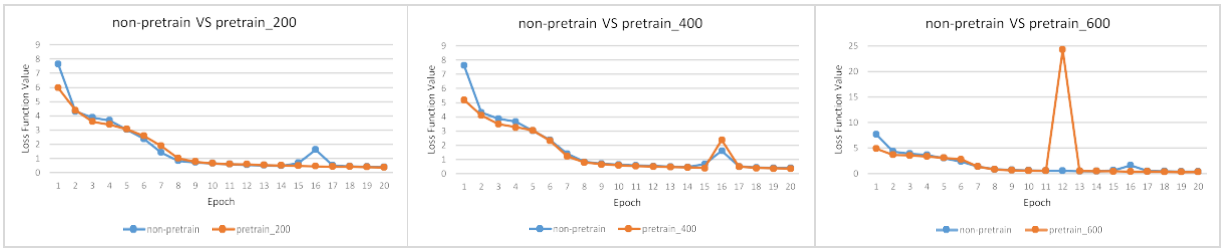


Fig. 8. Monitoring loss function values of pretrained R-SSD with different number of images and non-pretrained R-SSD.

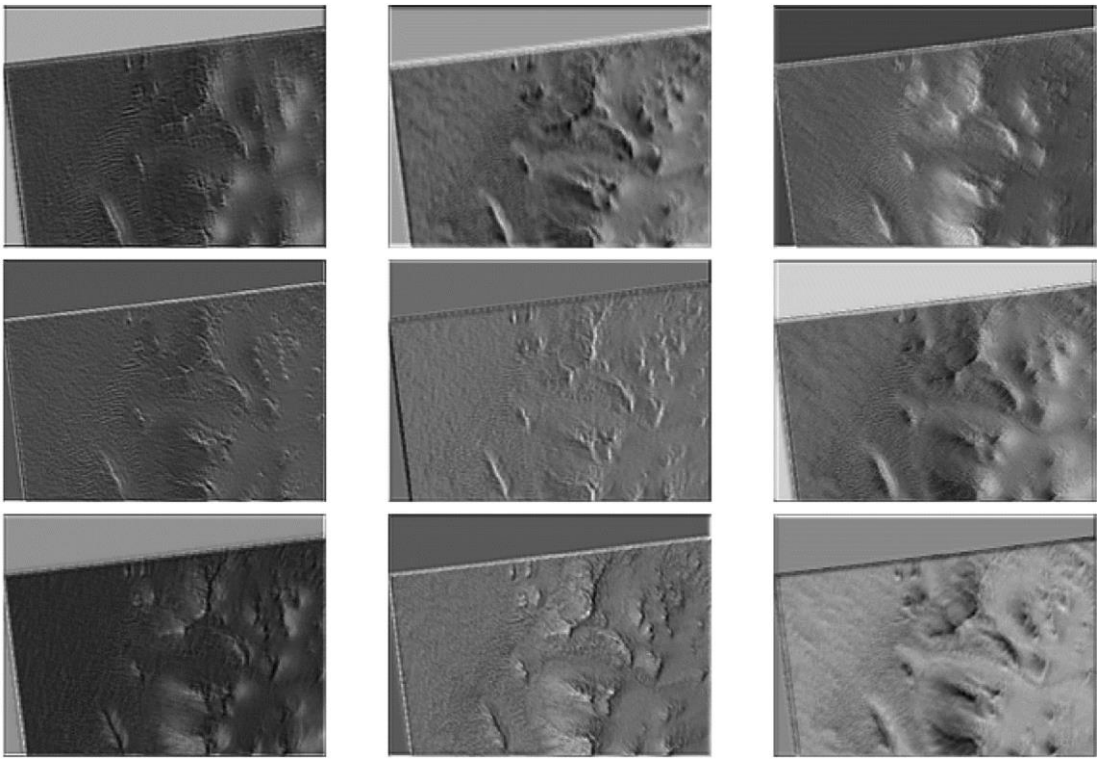


Fig. 9. Visualization of nine filters learned by the autoencoder after the pretraining process.

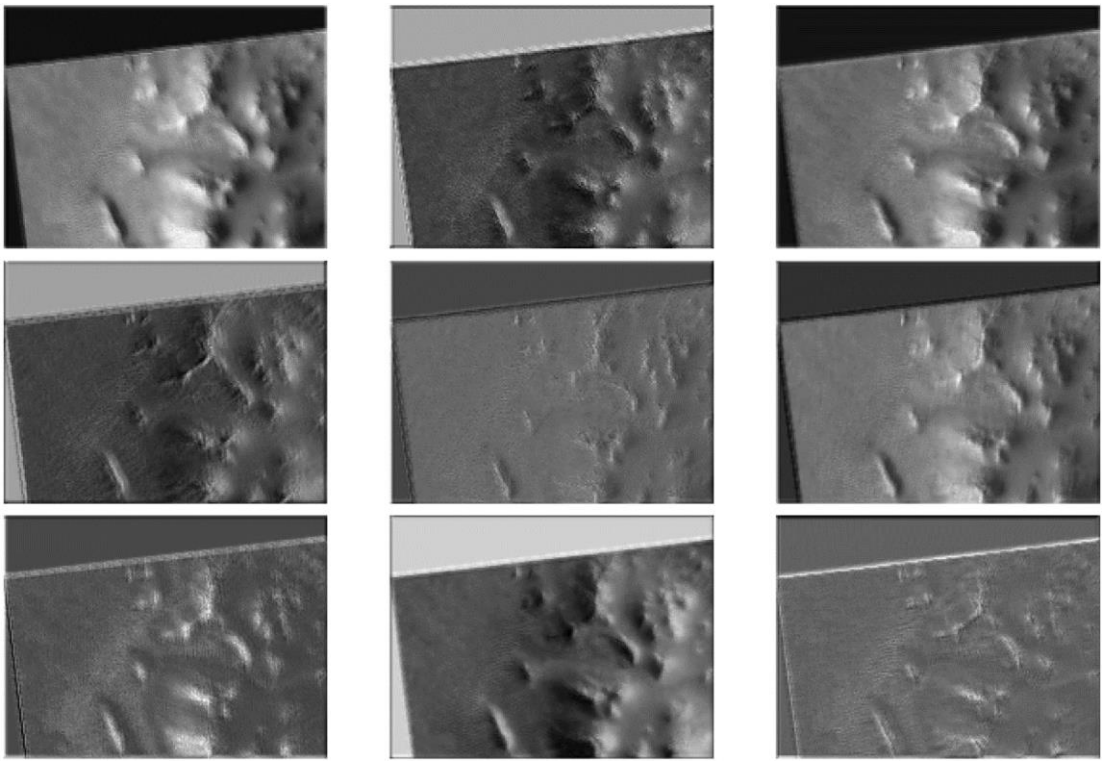


Fig. 10. Visualization of nine filters in the autoencoder with randomly generated model weights.

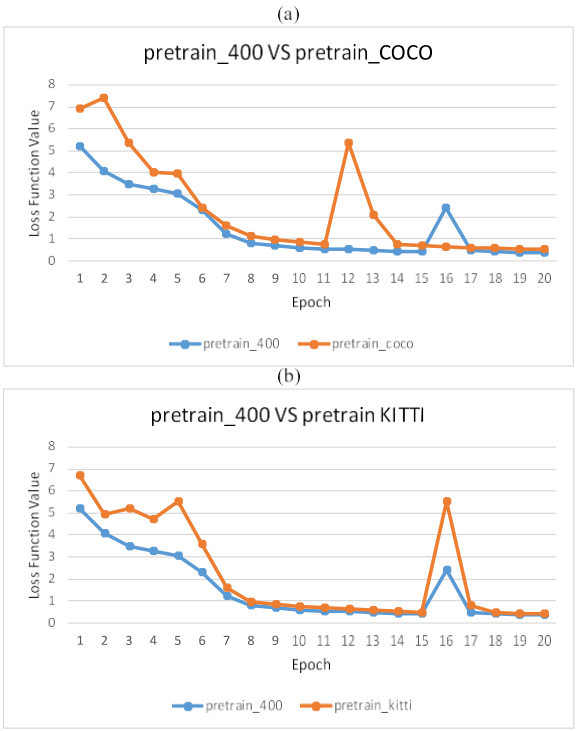


Fig. 11. Monitoring loss function values of R-SSD pretrained from different data sets.

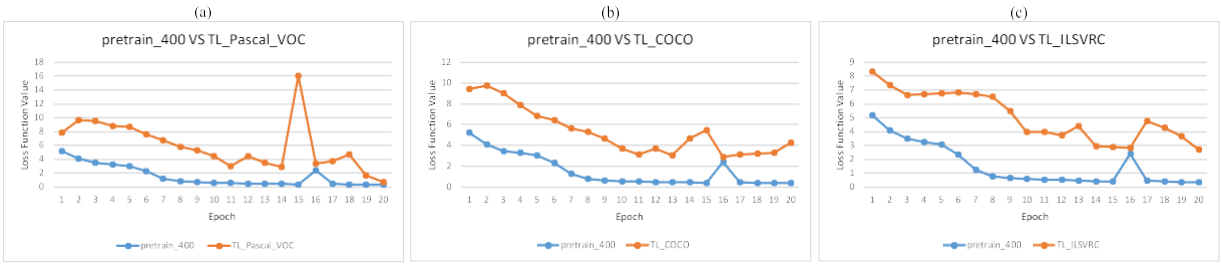


Fig. 12. Monitoring loss function values of R-SSD with autoencoder-based pretraining and transfer learning scheme.

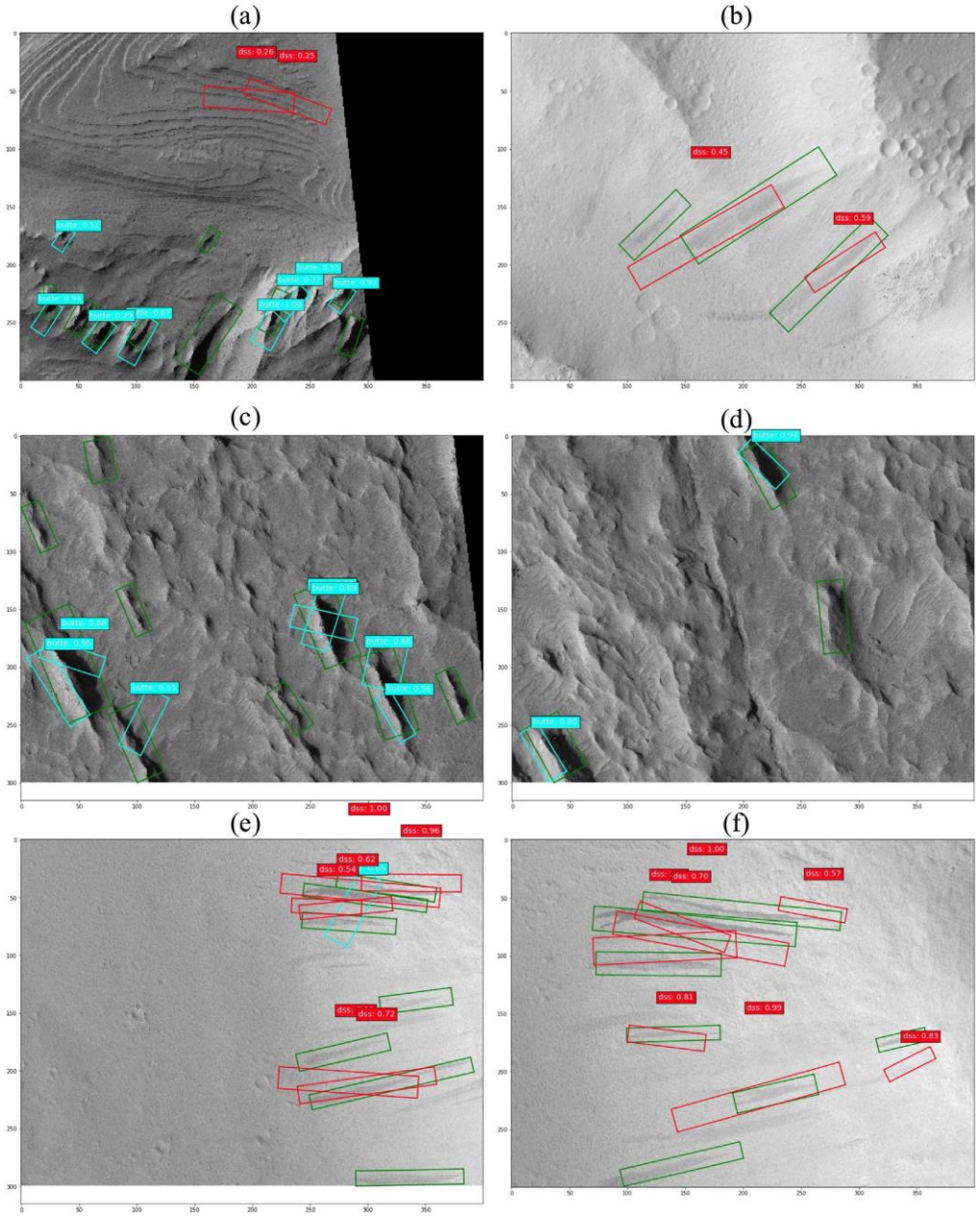


Fig. 13. Demonstration of typical detection results of R-SSD with pretrain_400: (a) and (b) are sampled from the testing set I; (c), (d), (e), and (f) are sampled from the testing set II.

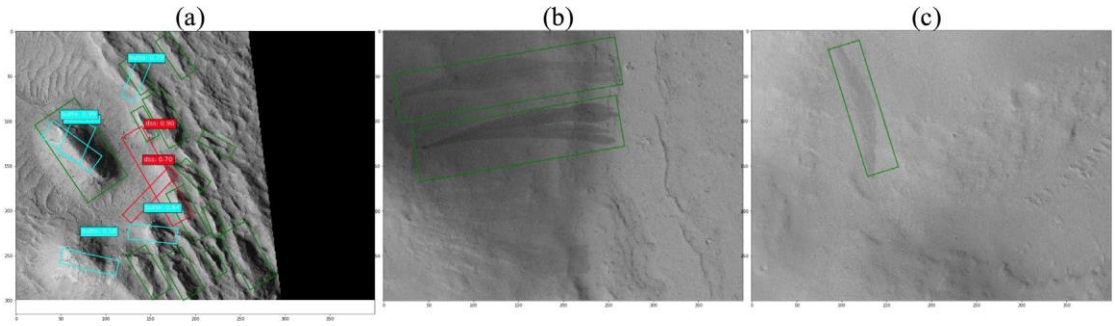


Fig. 14. Demonstration of some difficult cases that our system yields low-quality detections.

Table 1

Number of images and target objects in each data set.

Data set	No. of images	Total No. of buttes	Total No. of DSSs
training set	232	733	428
testing set I	10	33	26
testing set II	65	396	98

Table 2

Comparison results between different match strategies.

Match strategy	Model training time	AP on testing set I		AP on testing set II	
		Buttes	DSSs	Buttes	DSSs
MS	14198 s	0.85	0.631	0.535	0.194
MS1	343946 s	0.531	0.3	0.479	0.246
MS2	344185 s	0.503	0.391	0.435	0.154
MS3	14245 s	0.84	0.691	0.535	0.169
MS4	134311 s	0.48	0.191	0.49	0.186

Table 3

Comparison results between non-pretrained and pretrained R-SSD.

Model	AP on testing set I		AP on testing set II	
	Buttes	DSSs	Buttes	DSSs
non-pretrain	0.85	0.631	0.535	0.194
pretrain_200	0.912	0.416	0.548	0.331
pretrain_400	0.949	0.535	0.628	0.45
pretrain_600	0.8	0.701	0.841	0.298

Table 4

Detection results of R-SSD pretrained from different data sets.

Model	AP on testing set I		AP on testing set II	
	Buttes	DSSs	Buttes	DSSs
pretrain_400	0.949	0.535	0.628	0.45
pretrain_COCO	0.676	0.342	0.503	0.248
pretrain_KITTI	0.809	0.51	0.644	0.082

Table 5

Detection results of R-SSD with autoencoder-based pretraining and transfer learning scheme.

Model	AP on testing set I		AP on testing set II	
	Buttes	DSSs	Buttes	DSSs
pretrain_400	0.949	0.535	0.628	0.45
TL_Pascal_VOC	NA*	NA	0.036	0.027
TL_COCO	NA	NA	NA	NA
TL_ILSVRC	NA	NA	0.05	0.058

*NA means the corresponding model does not have any positive detections, so the AP cannot be calculated.

Table 6

Detection results of two hand-crafted feature-based object detection methods.

Model	Accuracy on testing set I		Accuracy on testing set II	
	Buttes	DSSs	Buttes	DSSs
morphology-based	9.1%	0	30.9%	1.4%
color-based	30%	0	44%	2.4%

Table 7

The computational complexity and model parameter size of R-SSD and benchmark methods.

Model	No. of MACs	No. of Params.
R-SSD	9.567G	54.67M
YOLOv2	4.266G	50.568M
YOLOv3	9.519G	61.626M
Faster-RCNN	133.971G	41.304M

Table 8

Detection results of R-SSD and benchmark deep learning-based object detection methods.

Model	AP on testing set I		AP on testing set II	
	Buttes	DSSs	Buttes	DSSs
R-SSD	0.85	0.631	0.535	0.194
YOLOv2	0.061	0.007	0.012	0.0014
YOLOv3	0.131	0.0001	0.18	0.0018
Faster-RCNN	0.418	0.406	0.175	0.103