Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8842227, 14 pages
https://doi.org/10.1155/2020/8842227

Research Article

Terrain Classification Algorithm for Lunar Rover Using a Deep Ensemble Network with High-Resolution Features and Interdependencies between Channels

Lanfeng Zhou, Ziwei Liu, and Wenfeng Wang

Shanghai Institute of Technology, Shanghai, China

Correspondence should be addressed to Lanfeng Zhou; lfzhou@sit.edu.cn and Wenfeng Wang; wangwenfeng@sit.edu.cn

Received 25 June 2020; Revised 18 August 2020; Accepted 3 September 2020; Published 14 October 2020

Academic Editor: Yin Zhang

For terrain classification tasks, previous methods used a single scale or single model to extract the features of the image, used high-to-low resolution networks to extract the features of the image, and used a network with no relationship between channels. These methods would lead to the inadequacy of the extracted features. Therefore, classification accuracy would reduce. The samples in terrain classification tasks are different from in other image classification tasks. The differences between samples in terrain classification tasks are subtler than other image-level classification tasks. And the colours of each sample in the terrain classification are similar. So we need to maintain the high resolution of features and establish the interdependencies between the channels to highlight the image features. This kind of networks can improve classification accuracy. To overcome these challenges, this paper presents a terrain classification algorithm for Lunar Rover by using a deep ensemble network. We optimize the activation function and the structure of the convolutional neural network to make it better to extract fine features of the images and infer the terrain category of the image. In particular, several contributions are made in this paper: establishing interdependencies between channels to highlight features and maintaining a high-resolution representation throughout the process to ensure the extraction of fine features. Multimodel collaborative judgment can help make up for the shortcomings in the design of the single model structure, make the model form a competitive relationship, and improve the accuracy. The overall classification accuracy of this method reaches 91.57% on our dataset, and the accuracy is higher on some terrains.

1. Introduction

Terrain classification is important in the driving process of a lunar rover, especially in complex terrain environments. There are two main directions for terrain classification. The first one is to classify the terrain by the vibration at which the rover through the ground. The second is to classify the terrain by the vehicle camera. The second method is more widely used than the first, because, in most situations, people need to predict the terrain in front of them.

The model proposed in this paper is based on the second direction. The base approach is to extract information from terrain images, such as spectra, colour, texture, and scale-invariant feature transform (SIFT) features [1–5]. The terrain can be accurately identified by that.

Although considerable research has been conducted on terrain classification in recent years, the bulk of this research is focused on man-made environments [6] or use more traditional algorithms for classification. Howard and Seraji [7] of the Jet Propulsion Laboratory did a lot of research on terrain description and terrain traversability estimation. Visual characteristics and fuzzy rules are used to estimate the traversability of the terrain. Firstly, the surrounding terrain images are obtained through vehicle cameras. The roughness, slope, discontinuity, hardness, and other information of the surrounding terrain are obtained through image analysis. Then, the type of terrain is judged according to the established fuzzy rules. This paper gives us clear classification rules and a normative reference. Iagnemma et al. [8] of the Massachusetts Institute of Technology

proposed an online terrain parameter estimation method. Lauro Ojeda of the University of Michigan did some new research on terrain classification and some terrain descriptions [9]. They used a fully connected neural network with only one hidden layer to classify the terrain. They divided the samples into five categories. They are gravel, grass, sand, and pavement dirt. The accuracy of this model reached 78.4%. He et al. [10] proposed a hierarchical classification approach. The Conditional Random Field (CRF) and the Bayesian Network (BN) are employed to incorporate prior knowledge, to facilitate SAR image classification. However, from DeepLab v3 [11] to DeepLab v3+ [12], the CRF block is replaced by complex neural networks. It means that neural networks can replace traditional machine learning methods in some tasks, and even neural networks will perform better.

In addition, with the continuous development of artificial intelligence, more and more intelligent algorithms, such as CNN and unsupervised learning, are used for terrain classification. Zeltner [13] used a deep convolutional neural network to implement a vision-based terrain classification. Park et al. [14] proposed a new classification network framework based on LSTM units and ensemble learning. Lu et al. [15] detected deep-sea images by using YOLO. Bai et al. [16] proposed an improved terrain classification method based on three-dimensional vibration information for terrain classification.

It is worth noting that most previous works focus on the extraction of salient features. The samples have significant differences between each other in those tasks. However, performances of the most previous models will be degraded when the differences between images are subtle.

Combined with the above analysis, a new terrain classification method based on the combination of convolutional neural network and ensemble learning is proposed. The main contributions of this paper are establishing interdependencies between the channels to highlight the image features and maintaining a high-resolution representation throughout the process to ensure the extraction of fine features. We get the channel descriptor by using global average pooling along the channel direction on feature maps after each convolution. This descriptor has a global receptive field. We take this descriptor as input to a fully connected network. The output has the same number of channels as input. The weights of this fully connected network are used to represent the interdependencies among various channels. The output is weighted channel-by-channel to previous features by multiplication to achieve a feature map with different channel weights. To achieve high-resolution features, we connect high-to-low resolution subnetworks in parallel. The information is exchanged in parallel multiresolution subnets. Information can be exchanged directly between the same resolutions. Information is exchanged from high-resolution to low-resolution by downsampling. Information is exchanged from low-resolution to high-resolution by upsampling. We discard low-resolution features because the differences between the samples are subtle. Meanwhile, a new activation function Mish [17] is used in the optimization of the model. Mish is a new activation function that is proposed by Diganta Misra. It performs better than ReLU on many datasets. Besides, we use a shallow neural network to integrate the output of each model to get the final results. Multiple models can be more effective to prevent overfitting than a single model. In addition, our method is easy to implement in practical applications. The remainder of this paper is organized as follows.

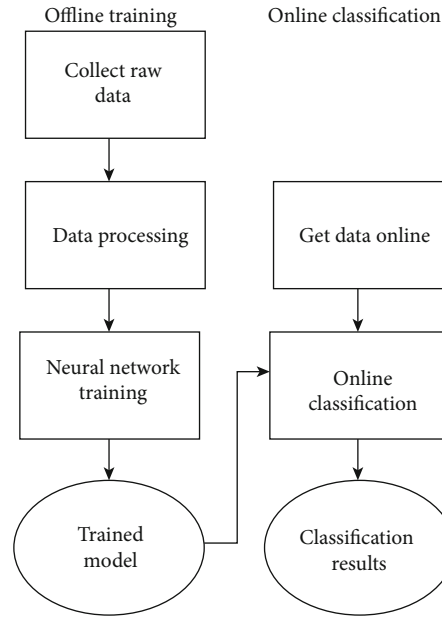


Figure 1: Schematic diagram of the algorithm flow framework.

In Materials and Methods, we review the previous methods, standards for terrain classification, and the theoretical basis of the convolutional neural network (CNN). We take these as the basis of our research. Meanwhile, we give our specific model. In Results and Discussion, we describe the experimental process of our model and the results of our experiments. In Conclusions, we summarize and analyse the experimental results.

2. Related Work

2.1. Terrain Classification. The main target of terrain classification is that we can quantify the ease-of-traversal of terrain by a mobile robot based on real-time measurements of terrain characteristics retrieved from vehicle cameras. Howard and Seraji [7] used a rule-based Fuzzy Traversability to classify the terrain. These characteristics include, but are not limited to slope, roughness, hardness, and discontinuity. The classification criteria of our experimental raw data are based on the above indicators.

2.2. Image Classification. In recent years, automatic classification techniques based on neural networks have been more and more. From 2012, on the competition of ImageNet [18], AlexNet [19] was proposed. On the ICLR2015, the VGG [20] was proposed. On the CVPR2018, Google proposed NasNet [21]; it was training on 500 GPUs. The accuracy of the top 5 and top 1 of the competition of ImageNet increased from 57.1% and 76.3% to 96.2% and 82.7%. It

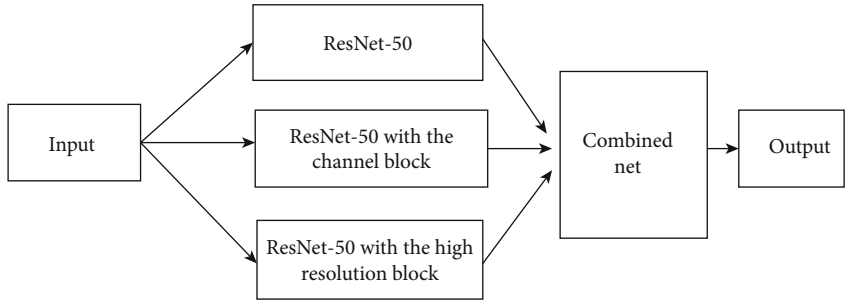


Figure 2: The structure of the ensemble network.

Table 1: A detailed description of the original ResNet-50 model.

conv1	7×7, 64, stride 2
	3×3, maxpool, stride2
conv2_x	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 256 \end{bmatrix} \times 4$
conv4_x	$\begin{bmatrix} 1 \times 12, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
conv5_x	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
fc1	Average pool, 1000-d fc, Softmax
fc2	4-d fc

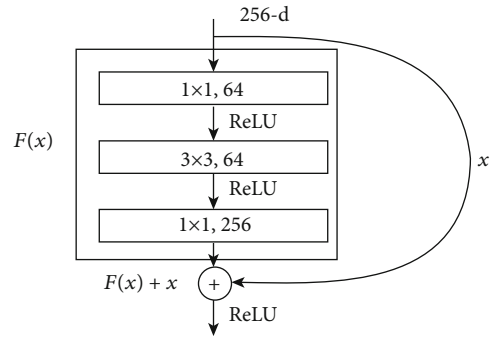


Figure 3: conv2_x structure.

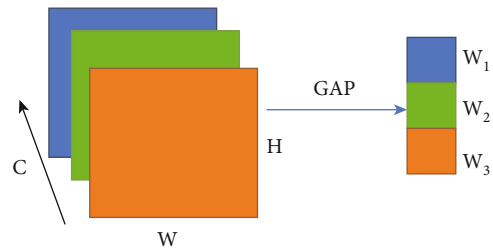


Figure 4: The process from the feature map to the initial descriptor of each channel.

can be seen that the accuracy of image classification based on convolutional neural networks is constantly improving.

2.3. Convolutional Neural Network. Neural networks were proposed by mimicking human brains. Convolutional neural networks are a special structure of neural networks, which are widely used in the field of computer vision. The image is divided into small regions in the same manner as the brain perceives the object. The features of each region are learned to classify the input image.

A simple neural network is a chained structure in which each layer is a function of the previous layer. [22, 23] The first layer:

$$H^{(1)} = g^{(1)}\left(W^{(1)T}x + b^{(1)}\right), \tag{1}$$

$H^{(1)}$ is the output of the first layer. $g^{(1)}$ is a nonlinear variation function. $W^{(1)}$ is the weight matrix, which the values we need to train. b is bias. x is an input layer. It is a vector.

The second layer:

$$H^{(2)} = g^{(2)}\left(W^{(2)T}H^{(1)} + b^{(2)}\right), \tag{2}$$

$H^{(1)}$ is the output of the first layer. So, we can express the output of n layer as:

$$H^{(n)} = g^{(n)}\left(W^{(n)T}H^{(n-1)} + b^{(n)}\right). \tag{3}$$

There is a theory that we can represent any arbitrary function by a neural network that has more than two layers. But according to experimental experience, training a deep network requires much fewer parameters than training a shallow network. The reason is that the low layers have already extracted the basic features, and the high layers only need to combine these basic features to get more complex

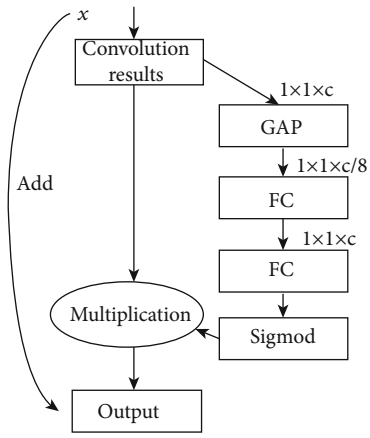


Figure 5: The structure of the fully connected network in the channel block.

Table 2: A detailed description of the ResNet-50 with channel block.

conv1	7×7, 64, stride 2	
	3×3 maxpool, stride 2	
conv2_x	$\begin{bmatrix} \text{conv}, 1 \times 1, 64 \\ \text{conv}, 3 \times 3, 64 \\ \text{conv}, 1 \times 1, 256 \\ fc, [32, 256] \end{bmatrix}$	$\times 3$
conv3_x	$\begin{bmatrix} \text{conv}, 1 \times 1, 128 \\ \text{conv}, 3 \times 3, 128 \\ \text{conv}, 1 \times 1, 512 \\ fc, [64, 512] \end{bmatrix}$	$\times 4$
conv4_x	$\begin{bmatrix} \text{conv}, 1 \times 1, 256 \\ \text{conv}, 3 \times 3, 256 \\ \text{conv}, 1 \times 1, 1024 \\ fc, [128, 1024] \end{bmatrix}$	$\times 6$
conv5_x	$\begin{bmatrix} \text{conv}, 1 \times 1, 512 \\ \text{conv}, 3 \times 3, 512 \\ \text{conv}, 1 \times 1, 2048 \\ fc, [256, 2048] \end{bmatrix}$	$\times 3$
fc1	Average pool, 1000-d fc, Softmax	
fc2	4-d fc	

features. It is similar to modularization in industrial production [24, 25].

3. Materials and Methods

3.1. Lunar Terrain Classification

3.1.1. Method Overview. The entire classification process is divided into two phases. The first part is the offline training.

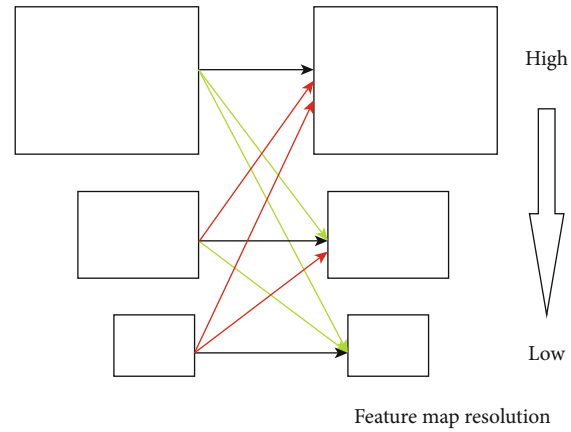


Figure 6: The process of exchanging information in three kinds of resolution in each channel.

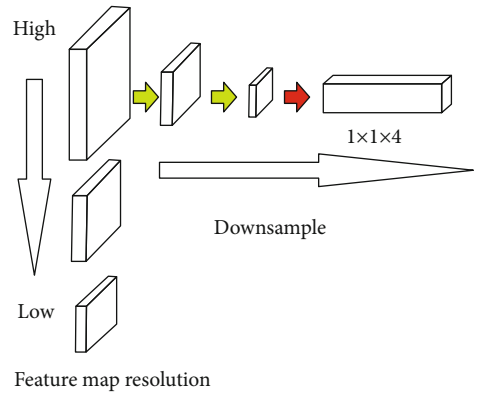


Figure 7: The process of constructing the classifier in three kinds of resolution.

The second part is the online classification. As shown in Figure 1, in the offline training part, we use more than 3,000 photos taken by Chang'e-3 as raw data. The data processing includes cleaning, labelling, dividing, and enhancing data. We train the model with our data. After a period of training, we get a trained model. In the online classification part, we get data online as inputs. Next, we use that trained model to classify the terrain. The trained model's outputs are the classification results. This is the whole process of the terrain classification algorithm.

3.2. Model. We use ResNet-50 [26] as a backbone and two functional blocks. One of the functional blocks is to establish the interdependencies between the channels. We call it the channel block. Another one is to maintain a high-resolution representation. We call it the high-resolution block. Our model includes 4 networks, original ResNet-50, ResNet-50 with the channel block, ResNet-50 with the high-resolution block, and a combined network. We divide samples into three groups and use three different models to train. Combined with three model results, the final results can be obtained. Ensemble learning is a technique that can alleviate overfitting problems. Figure 2 shows the structure of our ensemble network:

Table 3: A detailed description of the ResNet-50 with high-resolution block.

Resolution	Stage 1	Stage 2	Stage 3	Stage 4	Head
1	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times4\times1$	$\begin{bmatrix} 3\times3, 48 \\ 3\times3, 48 \end{bmatrix}\times4\times1$	$\begin{bmatrix} 3\times3, 48 \\ 3\times3, 48 \end{bmatrix}\times4\times4$	$\begin{bmatrix} 3\times3, 48 \\ 3\times3, 48 \end{bmatrix}\times4\times3$	Two 3×3 convolution for downsample
1/2		$\begin{bmatrix} 3\times3, 96 \\ 3\times3, 96 \end{bmatrix}\times4\times1$	$\begin{bmatrix} 3\times3, 96 \\ 3\times3, 96 \end{bmatrix}\times4\times4$	$\begin{bmatrix} 3\times3, 96 \\ 3\times3, 96 \end{bmatrix}\times4\times3$	
1/4			$\begin{bmatrix} 3\times3, 192 \\ 3\times3, 192 \end{bmatrix}\times4\times4$	$\begin{bmatrix} 3\times3, 192 \\ 3\times3, 192 \end{bmatrix}\times4\times3$	
1/8				$\begin{bmatrix} 3\times3, 384 \\ 3\times3, 384 \end{bmatrix}\times4\times3$	

The following Table 1 is a detailed description of the original ResNet-50 [26] model.

The shapes and operations with specific parameter settings of a residual building block are listed inside the brackets and the number of stacked blocks in a stage is presented outside. The fc1 and fc2 mean two fully connected layers [27]. The ReLU activation function is not shown for brevity.

The convolution formula is as follows:

$$p_{i,j} = f\left(\sum_{d=0}^{D-1}\sum_{m=0}^{F-1}\sum_{n=0}^{F-1} w_{d,m,n}x_{d,i+m,j+n} + w_b\right) \tag{4}$$

$p_{i,j}$ is the pixel value of row i and column j, D is the depth of the convolution kernel, F is the size of the convolution kernel, $w_{d,m,n}$ is the weight of the convolution kernel in row m, and column $n\,c$; w_b is the bias.

ReLU activation function is as follows:

$$f(x) = \max(0, x) \tag{5}$$

is the output of the previous layer.

The linear calculation is as follows:

$$f = Wx + b \tag{6}$$

W is the weight of the model obtained through training, b is the bias term, and x is the output of the previous layer.

The Softmax function is as follows:

$$\text{softmax}(x_1, x_2, \cdots x_n) = \frac{e^{x_i}}{\sum_{i=1}^{n} e^{x_i}} \tag{7}$$

n is the number of types, and the Softmax value is the probability of each type. Obviously, the sum of all Softmax values is 1. A Softmax function is used to generate a label distribution containing 4 categories.

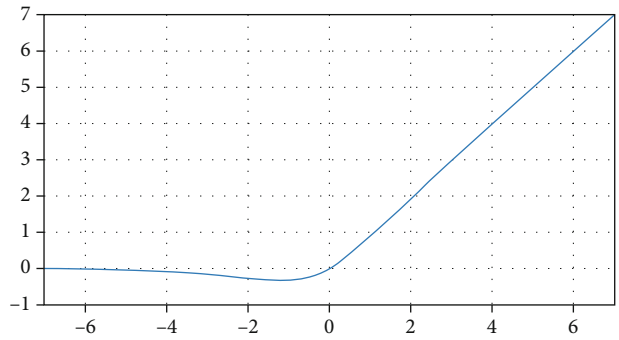


Figure 8: Mish function.

The cross-entropy:

$$\text{loss} = \sum_{i=1}^{n} p(x_i)\log\left(q(x_i)\right) \tag{8}$$

$p(x_i)$ is the true probability of x_i.$q(x_i)$ is the probability that is calculated by the model. The cross-entropy measures the distance between two distributions. The more similar the actual distribution to the predicted distribution, the smaller the value of the cross-entropy. Finally, we use Adam optimized gradient descent to solve this model. When the parameters converge, we can get the preliminary model.

3.3. Residual Block. To deal with the degradation problem, the residual block is proposed. The difference between ordinary neural networks is that the residual network has cross-layer connections, also called shortcut connections. A residual module is constructed by them. In a residual block, cross-layer connections generally span only two or three layers but do not exclude crossing more layers. The experimental results of the situation of crossing only one layer are not good. Figure 3 shows the conv2_x structure:

The residual block formula is as follows:

$$y = f(x, \{w_i\}) + x, \tag{9}$$

$f(x, \{w_i\})$is called residual mapping.

Table 4: A detailed description of the Combine Net.

Input	Layer 1	Layer 2	Result
ResNet-50			
ResNet-50 with channel block	Fully connected 512 neurons	Fully connected 4 neurons	Current terrain prediction results
ResNet-50 with high-resolution block			

The mapping between the l and l_1 layers is as follows:

$$a^{(l)} = f\left(a^{(l-1)}\right) + a^{(l-1)} = f\left(a^{(l-1)}\right) + f\left(a^{(l-2)}\right) + a^{(l-2)} = a^{(l_1)} + \sum_{i=l_1}^{l-1} f\left(a^{(i)}\right), \tag{10}$$

l, l_1 is any layer and $l > l_1$.

With the number of network layers deepening, the parameters of lower layers cannot be effectively updated in traditional networks. But in the residual block, we can solve it:

$$\frac{\partial \text{loss}}{\partial a^{(l_1)}} = \frac{\partial \text{loss}}{\partial a^{(l)}} + \frac{\partial \text{loss}}{\partial a^{(l)}} \frac{\partial}{\partial a^{(l_1)}} \sum_{i=l_1}^{l-1} f\left(a^{(i)}\right), \tag{11}$$

The gradient of the loss to a lower layer output is decomposed into two terms; the previous term $\partial \text{loss}/\partial a^{(l)}$ shows that error signals can propagate directly to lower layers without any intermediate weight matrix transformation. So the parameters of lower layers can be effectively updated. Residual connections make information flow more smoothly.

3.4. Channel Block. To establish the interdependencies between the channels, we use the channel block. Inspired by the 2017 ImageNet Challenge champion model [28], we find that the interdependencies between channels are useful to highlight features, especially for the classification model of samples with similar colours. We use the global average pooling(GAP) to get the descriptor of each channel. Figure 4 shows the process from the feature map to the initial descriptor of each channel. The number of channels is 3.

The global average pooling is as follows:

$$w_k = \sum_{i=0}^{W} \sum_{j=0}^{H} p_{i,j} / (W \times H), \tag{12}$$

w_k is the initial descriptor of the$k$ channel of the feature map. $p_{i,j}$ is the pixel value of row i and column j. W is the width of the feature map. H is the height of the feature map.

This descriptor has a global receptive field. Pixel context information is fully utilized. Getting this information is an important part of the picture-level classification.

We take the initial descriptors as the input of a fully connected network. The weights can represent the interdependencies between various channels by learning. The output is weighted channel-by-channel to previous features by multiplication to achieve a feature map with different channel weights. Figure 5 shows the structure of the fully connected network in the channel block.

Table 5: Rule base for Fuzzy Traversability Index.

Roughness	Discontinuity	Hardness	Type
Smooth	Small	Soft	Soft gravel
Smooth	Small	Hard	Compacted soil
Rough	Small	Hard	Compacted soil
Rough	Small	Soft	Soft gravel
	Large		Concave land
Rocky			Rocky terrain

FC is a fully connected layer. Sigmod is as the final activation function:

$$S(x) = \frac{1}{1 + e^{-x}}. \tag{13}$$

The output is a feature map with different channel weights.

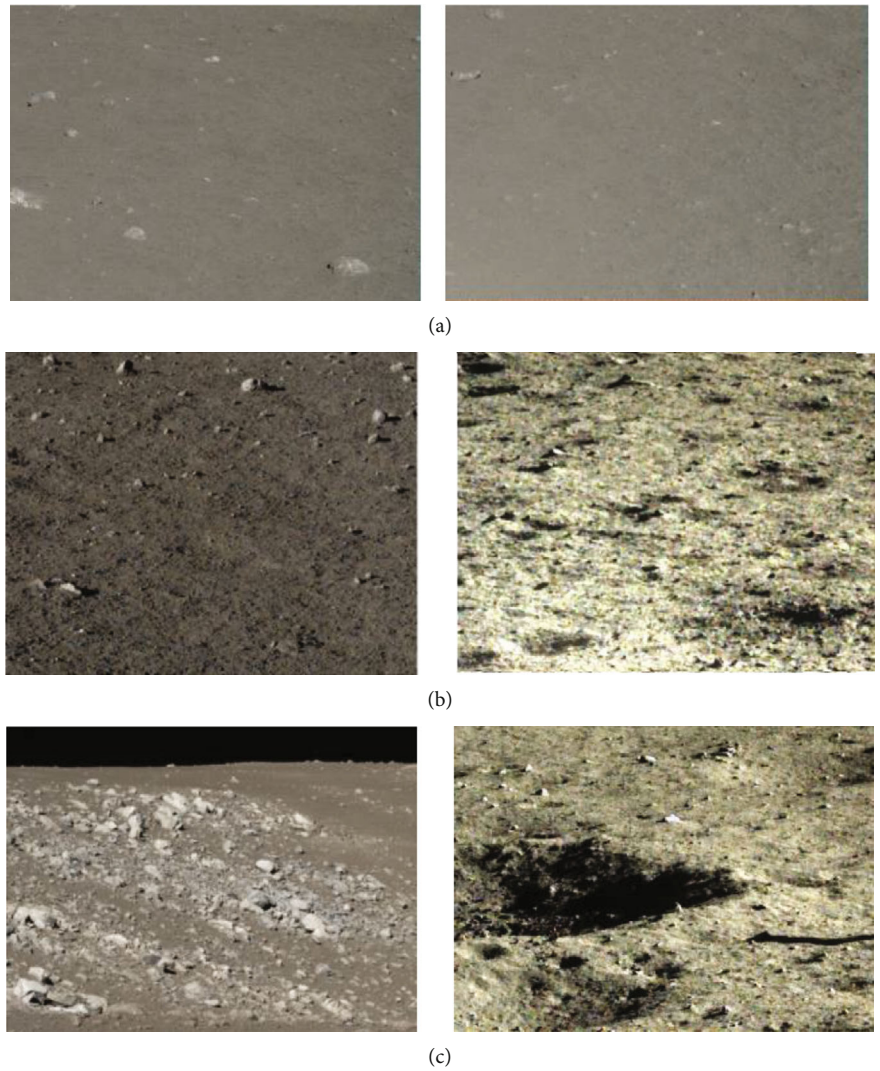
Table 2 is a detailed description of the ResNet-50 with channel block.

The shapes and operations with specific parameter settings of a residual building block are listed inside the brackets and the number of stacked blocks in a stage is presented outside. The inner brackets following by fc indicates the output dimension of the two fully connected layers in a channel block.

The fc1 and fc2 mean two fully connected layers.

3.5. High-Resolution Block. To get subtle features, we use the high-resolution block. Inspired by the research of Human Pose Estimation in CVPR2019 [29], we find that extracting high resolution can improve the accuracy of the classification model, especially for the classification model of samples with only subtle differences. We use convolution to downsampling when information is exchanged from high resolution to low resolution. On the other hand, we use transposed convolution [30] to upsampling when information is exchanged from low resolution to high resolution. Figure 6 shows the process of exchanging information in three kinds of resolution in each channel.

The black arrow means identity mapping. The red arrow means transposed convolution. The green arrow means ordinary convolution. We see the original convolution as a



(a) (b) (c)

Figure 9: Raw data from different terrain. (a) Soft gravel. (b) Compacted soil. (c) Rocky terrain (left) and concave land(right).

Table 6: The distribution of data.

Class	Soft gravel	Compacted soil	Rocky terrain	Concave land
Number	1043	2765	625	368

matrix operation. If the size of the feature map is 4×4, the size of the convolution kernel is 3×3, the stride is 1, and then the size of the output of the convolution operation is 2×2. We can unroll the feature map, the output, and the convolution kernel into vectors from left to right, top to bottom. The convolution can be represented as follows:

$$W = \begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{pmatrix} \tag{14}$$

The feature map can be represented as follows:

$$x^T = \left(p_{0,0}p_{0,1}p_{0,2}p_{0,3}p_{1,0}p_{1,1}p_{1,2}p_{1,3}p_{2,0}p_{2,1}p_{2,2}p_{2,3}p_{3,0}p_{3,1}p_{3,2}p_{3,3}\right). \tag{15}$$

The output can be represented as follows:

$$S^T = \left(y_{0,0}y_{0,1}y_{1,0}y_{1,1}\right). \tag{16}$$

The original convolution can be seen as a matrix operation as follows:

$$W \cdot x = S. \tag{17}$$

The transposed convolution is as follows:

$$x = W^T \cdot S. \tag{18}$$

W^T is the kernel of transposed convolution.

Since our samples of classification tasks are with only subtle differences, the low-resolution features are not important. When constructing the classifier, we only use the high-resolution features. Figure 7 shows the process of constructing the classifier in three kinds of resolution.

The green arrow means that we use a 2-stride 3×3 convolution outputting 256 channels to downsampling the high-resolution feature map. The red arrow means that we use a 1-stride 1×1 convolution outputting 4 channels and GAP to put the representations into the classifier. 1×1 convolution is very useful to make the number of channels consistent or make the number of the channels any value we want.

Table 3 is a detailed description of the ResNet-50 with high-resolution block.

3.6. Combine Net. The following is a detailed description of the Combine Net:

The first layer of Combine Net is a simple fully connected layer with 500 neurons, making a simple nonlinear transformation of the results from the previous models.

The second layer is also a simple fully connected layer with only four neurons, making a simple nonlinear transformation of the results from the previous layer.

The activation function of the first layer is a Mish layer.

The activation function of the second layer is a Softmax which gives the final probability of each type. Mish is a novel smooth and nonmonotonic neural activation function which can be defined as:

$$f(x) = x \cdot \tan h(\ln(1 + e^x)) \tag{19}$$

The graph of Mish is shown in Figure 8.

Table 4 is a detailed description of the Combine Net.

3.7. Data Processing

3.7.1. Data Augmentation Transformations. We use more than 3,000 photos taken by Chang'e-3 as raw data. After cleaning and cutting, we get 4,801 valid samples. The size of the samples is $784 \times 576 \times 3$. According to the rule-based Fuzzy Traversability, 4801 images are labelled. The rule-based definition of the Traversability Index in terms of terrain roughness, discontinuity, and hardness is summarized in Table 5.

According to Table 5, the 4,801 sample images were divided into four categories: soft gravel topography, compacted soil topography, rocky terrain, and concave land. Among them, the compacted soil topography is the optimal travel choice, and the soft gravel topography has a large slip ratio. Rocky terrain and concave terrain should be avoided choosing to travel as much as possible. The following four sets of images in Figure 9 show the four types of samples.

The distribution of data is shown in Table 6:

After that, we perform a series of transformations on the image as shown in Table 7 to keep the distribution of the data balanced and alleviate overfitting problems.

We transform rocky terrain, soft gravel concave, and land to increase data. We discard some compacted soil to keep the distribution of the data balanced. The number of each terrain samples is shown in Table 8 after the transformation.

3.7.2. Read the Large-Scale Datasets. We build a convolutional network model using the TensorFlow-GPU-1.12.0. The GPU is Titan. The CPU is Xeon6130. In the training process, we divide the data into three equal parts and train the

Table 7: Data augmentation transformations.

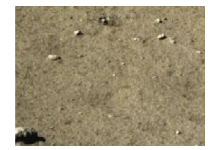
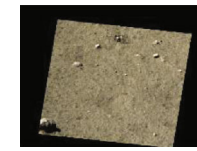
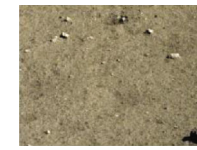
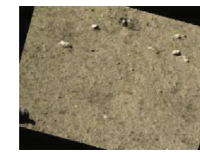
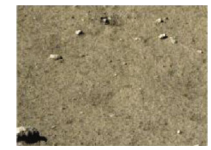
Transformation	Original	Image transform
Shear		
Flip right-left		
Rotate		
Scale		

Table 8: The number of images after the transformation.

Class	Soft gravel	Compacted soil	Rocky terrain	Concave land
Number	2086	2034	2187	2208

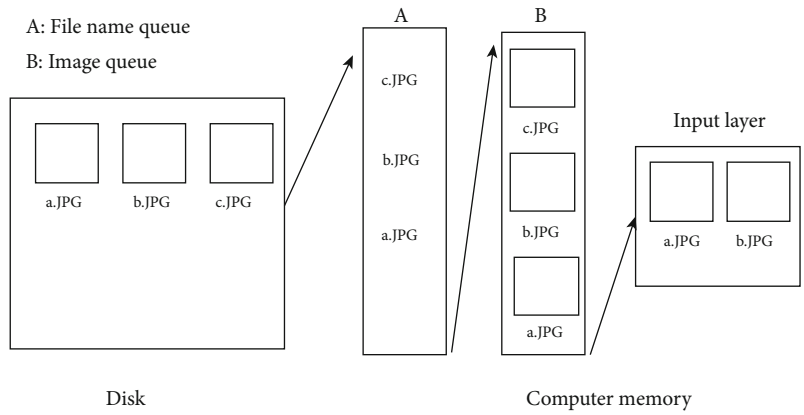


Figure 10: Picture reading process.

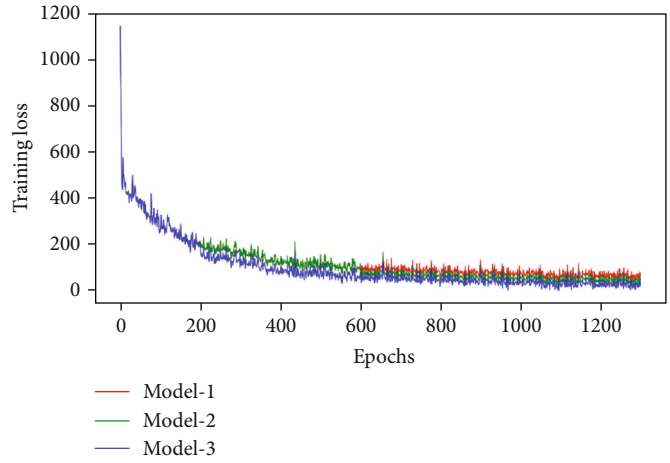


Figure 11: Loss values.

original ResNet-50 model, the ResNet-50 with the channel block model, and the ResNet-50 with the high-resolution model. The data are divided into the training set, the test set, and the verification set according to the ratio of 7 : 2 : 1. Every 16 pictures were used as a training batch. According to this, each batch is the input of the convolutional neural network for training, considering that I/O operations take a longer time than the encoding process and matrix operations. But we cannot put all the training data into computer memory because we do not have enough memory to store them. In order not to waste GPU resources, we use the multithreaded operation to train the model, as Figure 10 shows,

Specifically, a thread is used to continuously read the image name and path from the disk into the file name queue. A thread is used to continuously read images from the disk according to the name in the file name queue into the image queue. The input layer of the model can get images continuously from the image queue.

4. Results and Discussion

4.1. Experimental Process and Results. First, we read the data from the image queue.

Then, we divided the data into three equal parts. We put every batch of images into the corresponding network and train 1300 epochs. Model-1 is the original ResNet-50 model. Model-2 is the ResNet-50 with the channel block model. Model-3 is the ResNet-50 with the high-resolution model.

The loss values of the verification set are shown in Figure 11.

The accuracy of the verification set is shown in Figure 12.

According to Figures 11 and 12, we found that model-3 gets the best results. It means high-resolution features are important in the terrain classification task which the samples in the task only have subtle differences. Model-2 gets better results than model-1. It means establishing the interdependencies between the channels can highlight the features.

Finally, we use the idea of ensemble learning to fuse three models and then train a simple neural network based on the results of the three models. We take its output as the final output. This process is similar to a voting process, but the weights of each vote are nonlinear. We call the ensemble network as model-ensemble.

The accuracy of the validation set is shown in Figure 13.

Compared to a single model, the ensemble model is more accurate than a single model. Table 9 shows the results.

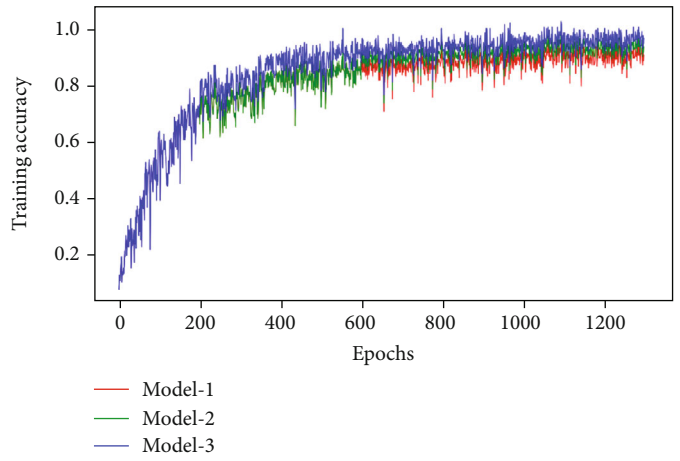


Figure 12: Accuracy values.

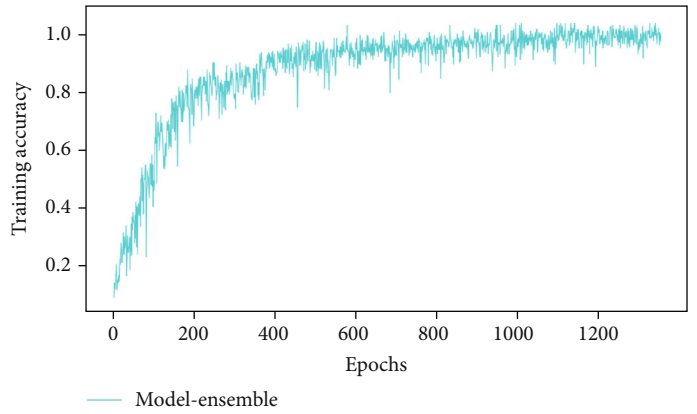


Figure 13: Accuracy of the ensemble model.

Table 9: Accuracy of the ensemble model.

Model	Train set	Test set	Val set
Model-1	92.61	91.07	89.75
Model-2	93.49	92.77	91.49
Model-3	94.11	93.16	91.54
Model-ensemble	94.31	93.24	91.57

Obviously, we see the result of model-ensemble is better than the result of model-3. The average accuracy is 91.57%. For the compacted soil terrain, the accuracy of our model is 95.37%. The accuracy of the soft gravel terrain is 93.95%, the accuracy of the rocky terrain is 89.13%, and the accuracy of the concave land terrain is 87.83%. The reason for the poor accuracy of the concave land terrain and the rocky terrain is that there are only 368 concave land images and 625 rocky terrain images in our sample. Even though we have enhanced the data, the results are still not good. Conversely, there are 2,765 compacted soil images. The accuracy of the compacted soil terrain is high. This indicates that the model requires a large number of samples to learn. In this situation, the model can learn to extract features and classify the terrain correctly.

From the above results, the imbalance of the data samples will cause the model to have different degrees of accuracy for different types of terrain. ROC curves and AUC values in Figure 14 intuitively express the ability of the model to recognise different terrains.

ROC curve of class 0-3 represents the ability of the model to recognize soft gravel, compacted soil, rocky terrain, and concave land. The average ROC curve is the average of the other 4 ROC curves. The larger the AUC values, the better the model. So the model is good at recognizing compacted soil. The reason is that there are more original data than others.

Considering the randomness of a single experiment, there are 20 random experiments to verify the model. The data is randomly selected from the data set. The results are shown in Figure 15. From the final results, the classification accuracy of the four types of terrain is 93.95%, 95.37%, 89.13%, and 87.83%, respectively. The results show that the classification accuracy of compacted soil and soft gravel is better than that of rocky terrain and concave land. Combined with the recognition accuracy of the whole dataset, it can be seen that the low accuracy may be caused by the difference in the amount of sample data. In the whole dataset, the amount of rocky terrain and concave land is smaller than that

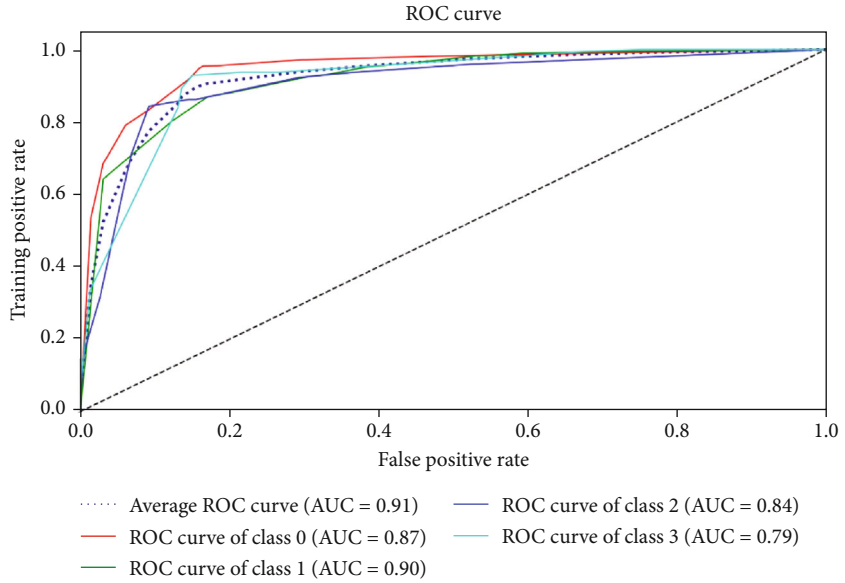


Figure 14: ROC curves and AUC values.

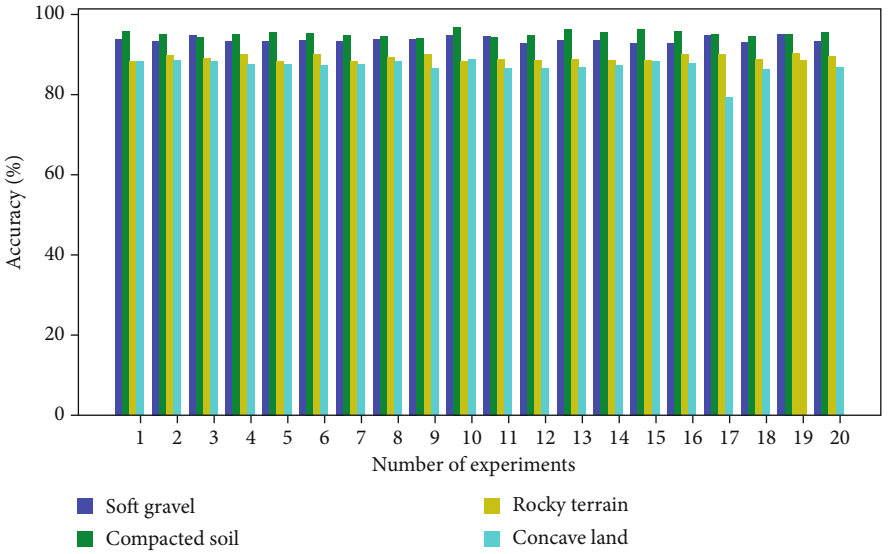


Figure 15: 20 experimental comparison results.

of soft gravel and compacted soil, which makes the learning accuracy inadequate.

Through the analysis of the confusion matrix of the experiment, 200 random data were classified into four types of terrain shown in Figure 16. It can be seen that the probability of rocky terrain and concave land being misclassified into wrong types of terrain is higher, which is also the reason for its low accuracy. We will be analyzing it in-depth in the follow-up study.

Since the geological composition of the moon is similar to that of the earth, we randomly searched for some scenes on the earth and used our model to classify. The results of the accuracy reached 90%. There are three samples of that random experiment in Table 10. The results mean that our model is valid to moon ground environment.

5. Conclusions

In this study, we proposed a method of the terrain classification algorithm for Lunar Rover by using a deep ensemble network with high-resolution features and interdependencies between channels. We use the original ResNet-50 model, the ResNet-50 with the channel block model, the ResNet-50 with the high-resolution model, and the Combine Net with a new activation function to solve terrain classification problems. To verify the algorithm, the experiment compares the performance of every single model on datasets and the performance of the deep ensemble network on datasets. The experimental results show that high-resolution features are important in the terrain classification task which the samples in the task only have subtle differences, establishing the

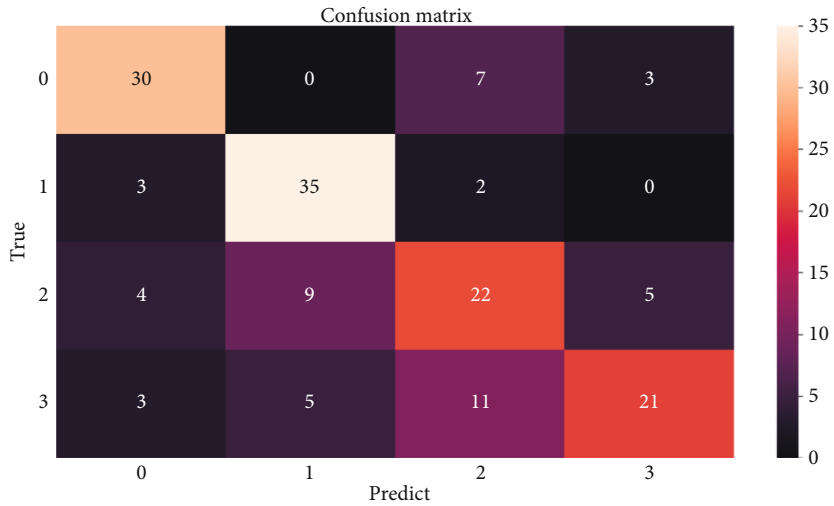


Figure 16: The results of classifying 200 random data in four types of terrain.

Table 10: The results of the random experiment.

Real scene	Classification result	True/false
	Soft gravel	True
	Compacted soil	True
	Rocky terrain	True

interdependencies between the channels can highlight the features, and multimodel collaborative judgment can help make up for the shortcomings in the design of the single model structure. Finally, the deep ensemble network reaches 91.57% average accuracy on our datasets.

Data Availability

You can get the raw data from here: http://moon.bao.ac.cn/

Conflicts of Interest

There is no conflict of interest regarding the publication of this paper.

Acknowledgments

The authors thank the research team for their support. This paper is funded by the National Natural Science Foundation (41671402).

References

[1] L. Semler and J. Furst, "Wavelet-based texture classification of tissues in computed tomography," in *18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*, pp. 265–270, Dublin, Ireland, 2005.

[2] G. Paschos, "Perceptually uniform color spaces for color texture analysis: an empirical evaluation," *IEEE Transactions on Image Processing*, vol. 10, no. 6, pp. 932–937, 2001.

[3] X. Liu and D. Wang, "Texture classification using spectral histograms," *IEEE Transactions on Image Processing*, vol. 12, no. 6, pp. 661–670, 2003.

[4] M. Pietikäinen, T. Mäenpää, and J. Viertola, *Color Texture Classification with Color Histograms and Local Binary Patterns*, IWTAS, New York, NY, USA, 2002.

[5] S. Zenker, E. E. Aksoy, and D. Goldschmidt, "Visual terrain classification for selecting energy efficient gaits of a hexapod robot," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 577–584, Wollongong, NSW, Australia, July 2013.

[6] J. Kruger, A. Rogg, and R. Gonzalez, "Estimating wheel slip of a planetary exploration rover via unsupervised machine learning," in *2019 IEEE Aerospace Conference*, pp. 1–8, Big Sky, MT, USA, March 2019.

[7] A. Howard and H. Seraji, "Vision-based terrain characterization and traversability assessment," *Journal of Robotic Systems*, vol. 18, no. 10, pp. 577–587, 2001.

[8] K. Iagnemma, H. Shibly, and S. Dubowsky, "On-Line Terrain Parameter Estimation for Planetary Rovers," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, pp. 3142–3147, Washington, DC, USA, 2002.

[9] L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen, "Terrain characterization and classification with a mobile robot," *Journal of Field Robotics*, vol. 23, no. 2, pp. 103–122, 2006.

[10] C. He, X. Liu, D. Feng, B. Shi, B. Luo, and M. Liao, "Hierarchical terrain classification based on multilayer Bayesian network and conditional random field," *Remote Sensing*, vol. 9, no. 1, p. 96, 2017.

[11] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, http://arxiv.org/abs/1706.05587.

[12] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, pp. 801–818, 2018.

[13] F. Zeltner, *Autonomous Terrain Classification through Unsupervised Learning*, University of Wurzburg, Master thesis, 2016.

[14] J. Park, K. Min, H. Kim, W. Lee, G. Cho, and K. Huh, "Road surface classification using a deep ensemble network with sensor feature selection," *Sensors*, vol. 18, no. 12, p. 4342, 2018.

[15] H. Lu, D. Wang, Y. Li et al., "CONet: a cognitive ocean network," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 90–96, 2019.

[16] C. Bai, J. Guo, and H. Zheng, "Three-dimensional vibration-based terrain classification for mobile robots," *IEEE Access*, vol. 7, pp. 63485–63492, 2019.

[17] D. Misra, "Mish: A Self Regularized Non-Monotonic Neural Activation Function," 2019, http://arxiv.org/abs/1908.08681.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami, FL, USA, June 2009.

[19] A. Krizhevsky, I. Sutskever, and G. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, NIPS. Curran Associates Inc., 2012.

[20] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014, http://arxiv.org/abs/1409.1556.

[21] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8697–8710, 2018.

[22] S. M. Ahn, "Deep learning architectures and applications," *Journal of Intelligence and Information Systems*, vol. 22, no. 2, pp. 127–142, 2016.

[23] L. Ogiela and M. R. Ogiela, "Beginnings of cognitive science," in *Advances in Cognitive Information Systems. Cognitive Systems Monographs, vol 17*pp. 1–18, Springer, Berlin, Heidelberg.

[24] F. Seide, G. Li, and D. Yu, "Conversational speech transcript using context-dependent deep neural networks," in *INTERSPEECH 2011 12th Annual Conference of the International Speech Communication Association*, Florence, Italy, August 2011.

[25] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8689*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., pp. 818–833, Springer, Cham, 2014.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[27] P. Simard, D. Steinkraus, and J. Platt, "Best practices for convolutional neural networks applied to visual document analysis," *ICDAR*, vol. 2, p. 958, 2003.

[28] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.

[29] S. Ke, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5693–5703, 2019.

[30] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2016, http://arxiv.org/abs/1603.07285.