

Received September 25, 2021, accepted October 20, 2021, date of publication October 27, 2021, date of current version November 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3123501

Image-Data-Driven Slope Stability Analysis for Preventing Landslides Using Deep Learning

BEHNAM AZMOON^{®1}, (Graduate Student Member, IEEE), AYNAZ BINIYAZ^{®1}, (Graduate Student Member, IEEE), ZHEN LIU^{1,2}, (Member, IEEE), AND YE SUN^{®3}, (Member, IEEE)

¹Department of Civil, Environmental, and Geospatial Engineering, Michigan Technological University, Houghton, MI 49931, USA

²Department of Geological and Mining Engineering and Science, Michigan Technological University, Houghton, MI 49931, USA

³Department of Mechanical and Aerospace Engineering and the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22903, USA

Corresponding author: Zhen Liu (zhenl@mtu.edu)

This work was supported in part by the National Science Foundation through the Civil, Mechanical and Manufacturing Innovation (CMMI) under Grant 1742656 and the Extreme Science and Engineering Discovery Environment (XSEDE) under Grant ACI-1548562.

ABSTRACT Landslides account for approximately 5% of natural disasters resulting in significant socioeconomic impacts. As a major infrastructure issue, slope stability has been traditionally analyzed with multiple deterministic and probabilistic methods to evaluate the stability of slopes or the probability of landslides. Geotechnical engineers tend to visit the sites of slopes, measure the geometry and soil properties, and use those traditional methods to analyze the slope stability and provide a factor of safety evaluation and recommendation. The fast-growing new technologies such as the internet of things and big data analytics provide new directions for natural hazard prevention. This study is the first to use deep learning as a new method for slope stability analysis for landslide prevention. A convolutional neural network was used to establish the model via transfer learning for processing simulated slope images. After training, our model can accurately predict the factor of safety of slopes for new slope images. Our proposed method was validated by comparing it with a classic limit equilibrium method, i.e., the simplified Bishop method, which is widely used in commercial programs for slope stability analysis. The comparison results showed that our proposed deep learning method outperformed the traditional method by decreasing the computation time by orders of magnitude without sacrificing accuracy. The results demonstrated the possibility and advantages of using deep learning as a new type of slope stability analysis method, including its ability to analyze raw image data directly, high level of automation, satisfactory accuracy, and short computing time, which will enable onsite evaluation for slope stability analysis. Thus, it facilitates fast in-situ decision-making for geotechnical applications and ensures the feasibility of using the internet of things and big data analytics for natural hazard prevention.

INDEX TERMS Convolutional neural networks, deep learning, landslides, natural hazards, resilience, slope stability analysis.

I. INTRODUCTION

The term 'landslide' refers to the movement of a mass of rock, debris, or earth down a slope [1]. Landslides have significant socio-economic impacts, cause losses of lives, and damage the environment. The estimated annual cost of landslides imposed on the U.S. economy is \$1.6 to \$3.2 billion,

The associate editor coordinating the review of this manuscript and approving it for publication was Young Jin Chun^(D).

and approximately 25-50 people are killed in the associated incidents every year [2]. In a broader picture, landslides constitute about 5% of natural disasters; this number is expected to increase due to the increase in population, unplanned urbanization, deforestation, and precipitation in some regions as a result of climate change [3], [4]. These facts render landslides a continuing concern and drive engineers to seek ways to improve the stability analysis of slopes.

Stability analysis of an earth slope quantifies its safety status. In theory, when the Factor of Safety (FS) obtained by stability analysis drops below one, a landslide occurs. The stability of the slope, or the probability of landslides, is mostly determined by its geometry and material properties. Traditionally, slopes to be analyzed are usually simplified into a 2D cross-section, which is analyzed together with the properties of the geomaterials constituting the slope. There are two major categories of methods: Limit Equilibrium Methods (LEMs) [5] and Strength Reduction Methods (SRMs) [6], which are the main deterministic approaches [7]. Among them, LEMs have been widely adopted to evaluate the FS of slopes due to their simplicity in estimating the landslide hazard. Various LEMs have been developed over decades, such as the Ordinary method of slices (Fellenius) [8], Simplified Bishop Method (SBM) [9], and Morgenstern-Price method [10]. The accuracy of these LEMs depends on their assumptions of the internal force distributions and shapes of slip surfaces [11]. Studies suggested that these methods can provide comparable performance to more refined methods in terms of calculating the average FS [12].

However, these traditional methods in slope stability analysis suffer from multiple issues such as the adoption of assumptions [13], difficulties in considering complicated field conditions [14], high computational costs, and high labor and skill requirements in modeling real-world problems. This is because assumptions and simplifications, which are hard to validate or assess, are usually needed due to the complex behavior of soils and rocks, unknown field conditions, uncertainty and variability in material properties, and limitations in the methods for modeling real-world problems [15]–[17]. In addition, the computing cost increases dramatically when more iterations or/and meshes are needed to allow for more complicated material properties and geometries [18]. These drawbacks limit the application of the traditional methods to simple field conditions and lead to errors caused by an oversimplified description of complex geosystems. In addition, the traditional methods are used case by case and cannot harness the knowledge that can be learned from existing cases. The recent advancements in deep learning [19] may open a door for providing new knowledge from "big data" as a significant direction and resource for future geotechnical engineering practices.

Thus, the goal of this study is to use Convolutional Neural Networks (CNNs) to develop an image-data-driven slope stability analysis method for preventing landslides using deep learning. This method is significantly faster, fully automated, and capable of predicting FS for a given slope while considering its soil property information, geometry, and critical slip circle. The scientific contributions of this study can be summarized as follows.

• To the best of our knowledge, this study is the first to explore the feasibility of using image-driven deep learning to estimate the *FS* of slopes for landslide prevention, which demonstrates the feasibility and advantages of

using deep learning as a new direction for landslide probability analysis.

- The above-discussed issues limit the traditional methods when analyzing complex real-world geosystems and jeopardize their accuracy. Our proposed method obviates the over-simplified assumptions and the time-consuming process of constructing a physics-based model. This is enabled by the automatic feature extraction capabilities of CNNs, leading to faster and more accurate slope stability analysis for landslide prevention.
- Considering the use of the fast-growing new technologies such as the internet of things and big data analytics, the results of this study may enable fast in-situ decisionmaking for solving traditional geotechnical engineering problems for natural hazard prevention.

The rest of the paper is organized as below: Section II briefly summarizes the literature about deep learning in geotechnical engineering applications and introduces the overview workflow of this study; Section III describes the proposed deep learning method for slope stability analysis; Section IV explains the big data of simulated slopes for deep learning; Sections V and VI present and discuss the results, respectively; and Section VII is the conclusion.

II. BACKGROUND

A. LITERATURE REVIEW OF CNNS IN GEOTECHNICAL ENGINEERING

Deep learning with CNNs has proven to be a successful tool to complement or replace current geotechnical engineering methods. One of the most significant advantages of CNNs is their automatic feature extraction capability [20]. In contrast to the traditional methods, CNN models work directly on raw image data to search for relationships between input and output, so there is no need to make assumptions or simplify the problem [21]. Taking slope stability analysis as an example, the input could be images of existing slopes, discrete elevation maps, or any image data containing the geometric, material, geological, geographical, or/and hydrological information of the slope, while the output could be the slope stability status such as the FS. The ability to extract knowledge from raw data is especially important due to the recent developments in sensor networks and technologies that promoted the generation of big data [22]. These changes in data generations contributed to the recent successes of neural networks in many areas, such as natural language processing and computer vision [23]. However, in traditional engineering applications such as slope stability analysis, it is still hard to utilize such world-changing advances in artificial intelligence to take advantage of the massive volumes of domain-specific data for stability analysis of geo-systems [24].

Though CNNs have not been explored extensively for geotechnical applications, the precursor of CNNs, i.e., traditional Artificial Neural Networks (ANNs), have been applied to virtually most types of geotechnical problems. In slope stability analysis, ANNs have been studied extensively, e.g., slope movements and landslide monitoring [25], analysis of rainfall-induced landslides [26], and landslide susceptibility mapping [27]. However, ANNs were unsuccessful in convincing most engineers of their ability to replace or even complement conventional methods [28]. One major reason might be their limited abilities to extract complicated features or insufficient computational capacities to analyze complex problems.

A noteworthy improvement of CNNs over ANNs was CNNs' ability to encode and extract image features into the architecture while reducing model parameters in the process. This enabled more highly automated solutions to more complicated tasks, e.g., analyzing images directly with affordable computing resources [29]. Such improvements prompted researchers to use CNNs in geotechnical applications in the recent few years. For example, CNNs have been used on drone-based thermal images for sinkhole detection [30], in tunneling to replace visual inspection [31], and in the safety analysis of retaining walls [32]. Especially, progress has been made in applying CNNs to a landslide-related topic: landslide susceptibility mapping. The published studies were mostly devoted to using CNNs for generating landslide susceptibility maps [33], [34], which proved CNNs could outperform other methods as a promising tool in landslide susceptibility mapping. Despite these pioneering efforts, deep learning with CNNs has been rarely utilized for geomaterials and geosystems, especially in landslide and stability analysis. This study is the first to use deep learning with CNNs for slope stability analysis with the hope of exploring a new direction for geotechnical applications.

B. WORKFLOW OF THIS STUDY

The workflow adopted for both conducting and reporting the study is presented in Fig. 1. The entire study is comprised of two main sections: developing the method and preparing the data for training and testing.

The method section details the theoretical foundation of establishing the deep learning model, whereas the data section covers the data preparation and analysis. In the method section, the reasons for choosing the deep learning framework and network architecture were discussed. Then, the theoretical basis and the procedure of applying multiclass classification to slope stability analysis were outlined with special attention to the mathematical background. At last, different types of errors and their causes were introduced.

In the data section, the FS was used to propose a data labeling method that is needed for the intended classification task in deep learning. Next, computer code that automated FScalculations with SBM, including the equations and assumptions of this LEM, was developed. Then, soil properties and random geometries were added so that data for deep learning can be obtained for slope stability analysis of slope images containing such information. The accuracy of our developed model was evaluated and validated using a commercial software package, Slide2, which is a 2D slope stability analysis program developed by RocScience. To enable the advances of



FIGURE 1. Workflow of the study.

deep learning, a large dataset including 11,480 simulated twodimensional simulated slope images were generated. These images were then preprocessed and labeled for training and analysis. Finally, pretreatments, including histogram equalization, and resizing were performed before dividing images into two sets for the later training and validation.

In the analysis section, the solver and the hyperparameters used for controlling the learning process were defined first. Using hyperparameters, the CNN was trained with the training set of images. After training is completed, the accuracy of the model was tested on a separate, completely independent set of images (validation/testing set). The testing results were compared against the *FS* prediction results of the SBM to evaluate the performance of the new method in terms of accuracy and computing time.

III. DEEP LEARNING FOR SLOPE STABILITY ANALYSIS A. ARCHITECTURE OF THE PROPOSED DEEP LEARNING METHOD

Advancements in machine learning and computer technology represented by CNNs enabled researchers to utilize automatic learning instead of hand-crafted feature extraction [35]. This is key in enabling direct analysis of images for stability assessments. Firstly, the architecture of the CNN used in this study is concisely discussed in this section.

The architecture of CNN comprises two main stages: a feature extraction stage and a classification stage, as shown in Fig. 2. In the first stage, the convolution layer is used to find the local conjunction of features from the previous layer. The convolution layer functions as filters working on arrays of image pixels as patches to obtain feature maps. Next, the image processed with a convolution layer passes through the Rectified Linear Unit (ReLU) to reduce nonlinearity and suppress overfitting, which is a typical problem in deep learning. Overfitting occurs when a model fits the noise instead of learning the underlying information and relationships in the



FIGURE 2. Architecture of the network.

training data, and consequently, fails to perform well on the testing data. The pooling operation is then used to reduce the size of the image while preserving its important characteristics. Pooling creates down-sampled feature maps or summarized versions of the features and reduces the number of parameters and operations in the network, and consequently, improves efficiency. Then, a Local Response Normalization (LRN) layer is used. Inspired by lateral inhibition in real neurons, LRN is a normalization scheme that aids generalization by creating competition between neuron outputs for big activities [36]. These steps are repeated multiple times to obtain the final feature map in the feature extraction stage. In feature extraction, images are broken down into features and analyzed independently.

In the second stage, the results from the first stage are fed into the fully connected layers (InnedProduct). Inner-Product layer determines the correlation between the position of features in an image and a particular class. The dropout technique is also used to prevent co-adaptation and overfitting issues [37] by temporarily removing units and their connections from the network during training [38]. The ultimate goal of the classification stage is to obtain a vector that has the same number of elements as that of classes. Each element in this vector specifies the probability that the image belongs to a predefined class. A Softmax layer was then employed to calculate the probability of an image belonging to each class. Fig. 2 demonstrates the hierarchy of layers in the model used in this study.

In this paper, a Python binding of Convolutional Architecture for Fast Feature Embedding (Caffe) developed and maintained by Berkeley Vision and Learning Center (BVLC) was selected as the deep learning framework. Caffe is a fully open-source framework that facilitates state-of-the-art deep learning with an extensive library of pre-trained reference models. Moreover, Caffe includes a C++ based implementation, bindings to Python/Numpy and MATLAB, and off-theshelf reference models [39]. Also, it features the separation of implementation and network definition and a high deployment speed. These advantages prompted us to select Caffe in this study.

B. THEORETICAL BASIS

This subsection discusses the theoretical basis of the application of deep learning in stability analysis. Especially, answers were sought for two questions: "Why deep learning with CNNs can be used to conduct stability analysis?" and "When deep learning results can be different from the true solution?"

Regarding the first questions, there are different ways to employ deep learning for meeting the goal of stability analysis, i.e., obtaining the FS of slopes. Two typical ways stem from the classification and regression abilities of deep learning. This study aims to utilize the potential of deep learning in image classification for obtaining the FS of slopes. Image classification used to be a challenging task for automated systems due to viewpoint-dependent object variability and high in-class variability of having many object types [40]. However, improvements in multiple layers of nonlinear information processing, GPUs, and large data sets enabled deep learning with CNNs to revolutionize image classification [41], [42]. The state-of-the-art CNNs can achieve superhuman classification skills, which inspired and formed the principal hypothesis of this study.

To use the classification ability of deep learning for slope stability analysis, a CNN needs to be employed to predict the category that a slope belongs to based on its FS. In theory, an infinite number of categories can be adopted to predict FS values in a continuous axis. However, multiclass classifications in deep learning are usually associated with a finite number of categories. Accordingly, slopes with FS values in different ranges are grouped into different categories, i.e., binned. For example, if slopes with FS values between any two neighboring digits in the first decimal place, e.g., between 1.2 and 1.3, are grouped into individual categories, then the classification result can reach FS predictions with one decimal place precision. Table 1 shows the different ranges of FS as well as their associated labels and categories that were adopted in this study. As listed, 1.5 was selected as a common recommendation for designing permanent slopes under static conditions (the ninth category). This threshold represents the "safe" condition of a slope despite the uncertainties involved in the FS calculation process. The same

TABLE 1. Ranges of FSs and their associated categories and labels.

Category	Range of FS	Label
First	Less than 0.8	0
Second	0.8-0.9	1
Third	0.9-1.0	2
Fourth	1.0-1.1	3
Fifth	1.1-1.2	4
Sixth	1.2-1.3	5
Seventh	1.3-1.4	6
Eighth	1.4-1.5	7
Ninth	Greater than 1.5	8

uncertainties prompted the use of 0.8 as a threshold for "fail" conditions (the first category). Moreover, it is widely believed that one decimal place precision is sufficient in slope stability analysis. Therefore, 0.1 increments were used to divide the FS values between 0.8 and 1.5 into seven categories (or called classes). Labels represent the unique identity of the categories, which are essential to the multiclass classification in deep learning.

Mathematically, the goal of deep learning is to find a mapping f from the input \mathbf{x} to the output \mathbf{y} : $\mathbf{y} = f(\mathbf{x} | \mathbf{w})$, in which f is a deep neural network characterized by its weights, \mathbf{w} . The input \mathbf{x} in this application is the image data of slopes. Each image can be viewed as an array of numbers corresponding to the pixel values. The output \mathbf{y} is related to the above labels and appears as a vector/array consisting of k numbers in a multiclass classification problem with k categories, in which any given element y_i in this vector is

$$y_i = \begin{cases} 1 & \text{if image belongs to this class} \\ 0 & \text{otherwise} \end{cases}$$
(1)

One key component in the use of deep learning for classification tasks is the last layer in the network. For simplicity, the mathematical operations occurring as the input moves from the first layer to the second last layer of the network can also be viewed as a mathematical function: $\mathbf{z} = \mathbf{z} (\mathbf{x} | \mathbf{w})$. \mathbf{z} is a vector that has k elements. In the last layer, a Softmax function is usually adopted to calculate the probability of the image belonging to the different categories in a multiclass classification problem. Therefore, this function takes \mathbf{z} as input and normalizes the vector into a probability distribution consisting of k probabilities as (2),

$$p_{i} = \frac{e^{z_{i}}}{\sum_{i=1}^{k} e^{z_{j}}}$$
(2)

where p_i is the probability that an image belongs to the *i* th category.

Deep learning is conducted to minimize the loss in the learning process. In multiclass classification with deep learning, this loss function is usually constructed with the cross-entropy:

$$J(\mathbf{w}) = -\sum_{i=1}^{k} [y_i \ln(p_i)].$$
 (3)

Deep learning typically comprises training and testing. In training, image data that is correctly labeled is fed as the input and output. Then the above loss function is minimized via an optimization process in the form of iterations to obtain a trained network represented by w. That process maximizes the overall probability that the predictions, i.e., predicted categories, equal the true classifications defined in the labels.

The second question is also important because it helps us understand errors and their causes when using deep learning for stability analysis. One type of error occurs in the above training process due to the fact that it is usually impossible to find the global minimum in the optimization, and overfitting/underfitting is hard to avoid. This type of error is relatively well understood and can be minimized with better CNNs and training techniques. This other type of error has not been well discussed in engineering and needs to be investigated considering the nature of the slope stability analysis. This type of error comes from sampling or the selection of data. To understand this error, Bayes's theorem is recalled:

$$p(A_i|B) = \frac{p(B|A_i) \cdot p(A_i)}{p(B)}$$
(4)

In the context of slope stability analysis, p(B) is the overall probability that the trained model can classify all images correctly and $p(A_i|B)$ is the probability that images in the *i* th category appear in the training data pertaining to *B*. In a classification problem with *k* categories, p(B) can be obtained as

$$p(B) = \sum_{i=1}^{k} [p(B|A_i) \cdot p(A_i)]$$
(5)

where $p(A_i)$ is the probability that images in the *i* th category appear in all the possible data, and $p(B|A_i)$ is the probability that the trained model with a classification ability of *B* can classify images belonging to the *i* th category. The Bayes's theorem is reformulated as follows to understand the second type of error:

$$\frac{p(B|A_i)}{p(B)} = \frac{p(A_i|B)}{p(A_i)}.$$
(6)

This equation indicates that the ratio between the probability that a trained model can classify one category of image data correctly and the overall accuracy of the trained model, i.e., left-hand side of (6), equals the ratio between the probability that this category of images appears in the training data and that in all the available data, i.e., righthand side of (6). If we aim at developing a trained model that can analyze all the slopes on Earth, then the available data should include image data covering representative slopes on this planet whose percentages reflect the reality. But for training a network, we can only use a small portion of the data from the available sources, leading to a difference between $p(A_i | B)$ and $p(A_i)$. $p(A_i)$ represents the percentage of the data in the *i* th category that is available in the place of applications, e.g., the Earth, which is also impossible to obtain in this application, whereas $p(A_i | B)$ can be controlled when selecting data for training. Therefore, according to (6), we can improve the performance of a CNN in classifying data with specific *FS* values by adding more data with *FS* values in the corresponding category. This is meaningful in the slope stability analysis because those slopes with *FS* values close to critical *FS* values, for example, *FS* = 1, are usually of more interest.

Despite the above theory, this study is mostly restricted to the first type of error, considering that the focus of this study is to show the feasibility and potential of deep learning in slope stability analysis. However, the second type of error will also be briefly discussed using the above theory. This is because the study adopted balanced datasets for training and testing, i.e., nine categories with the same number of images in each, to ensure that the prediction accuracies of the nine categories will not be much different from each other. A resampling processing that is required to obtain the balanced datasets from the original unbalanced ones will change the probability that a sample belongs to a certain category, i.e., $p(A_i | B)$ and $p(A_i)$, and consequently introduce the second type of error.

IV. DATA

A. TWO-DIMENSIONAL SLOPE DATA

In slope stability analysis of uniform slopes, 2D crosssections are normally used to determine the FS due to their simplicity and low cost. Although many studies have suggested that the 2D analysis tends to produce a more conservative estimation of the FS [43], this difference in FS is less than 10% for homogeneous material and in the absence of a significant load on the surface area of the slope, and thus is an effective way and widely used in slope stability analysis. Additionally, the accuracy of the 2D analysis depends on choosing the most critical cross-section within the sliding mass [44]. Figure 3 demonstrates the use of 2D analysis for a 3D slope failure. Figure 3(a) shows the Hillshade map of the area with a polygon denoting the landslide deposits. Figures 3(b) and 3(c) show the contoured topographic map of the area before (in 1954) and after (in 2014) the landslide, which is the 15th biggest filtered landslide that occurred in 1972, respectively. The red lines in these figures show the locations of the critical cross-sections for these slopes. These cross-sections are then employed to construct 2D models. One example of a 2D cross-section for slope stability analysis is shown in Fig. 3(d) [48]. After successfully generating 2D models, the slope stability can then be analyzed by the traditional methods. For example, Fig. 3(e) and 3(f) are typical analysis results using different commercial software.

Therefore, in this study, we used 2D cross-section slope image data for model training. To enable fast deep learning with big data, image data generated by the computer was used to serve the goal of proof of concept and to assess the possibility and performance of the new method more quantitatively and in a well-controlled setting.

B. DATA LABELING

The SBM [9] was selected as the slope stability analysis method for both labeling the image data and evaluating the performance of deep learning. The SBM adopts a few assumptions that can affect its accuracy. First, the SBM assumes a constant FS along the slip surface, while finite element analysis revealed that the FS varies considerably along the slip surface. Second, this method solely considers the interslice normal (horizontal) forces and neglects interslice shear forces. However, the difference in the average FS values



FIGURE 3. 3D to 2D slope data for slope stability analysis. (a) Hillshade map of the 15th biggest filtered landslide in 1972 from the Statewide Landslide Information Database for Oregon (SLIDO); (b) Same location using contour plot in 1954 before the landslide in (a) (scale: 63k); (c) Contour plot of the same location in 2014 after the landslide (scale: 24k); (d) 2D slope analysis using tool STB2010 [49]; (e) finite element analysis result of a 2D slope using Comsol; (f) 3D to 2D cross-section schematic diagram for slope stability analysis [48].



FIGURE 4. Algorithm flow.

calculated with SRM via the finite element method and SBM is less than five percent [12].

SBM calculates the *FS* as the total of the resisting forces divided by the total of the driving forces:

$$FS = \frac{\sum \left\{ \left(c' \Delta x + (w - u \Delta x) \tan \varphi' \right) \frac{1}{M_{\alpha}} \right\}}{\sum w \sin \alpha}$$
(7)

where α is the angle between the potential failure arc and the horizontal at the midpoint of the slice, w is the weight of the slice, c' is the cohesion, φ' is the angle of internal friction, Δx is the width of the slice, u is the pore pressure, and M_{α} can be calculated as

$$M_{\alpha} = \cos \alpha + \frac{\sin \alpha \tan \varphi'}{FS}.$$
 (8)

It is noted that M_{α} in (8) is a function of *FS*, while *FS* is also a function of M_{α} . Thus, an iterative procedure is needed for the solution. The implementation of the SBM with computer code to calculate *FS* for each slip circle in this study is illustrated in Fig. 4. The input for the SBM included the width of the slices, pore water pressure, soil properties, and

two-column matrices containing weights and α values for all the slices. The driving force of each slice was obtained by calculating the element-wise multiplication (Hadamard product) of sin α and w. The summation of all the elements of this resultant matrix was the denominator in (7) and remained constant for all iterations. Two initial guesses were needed to start the iterative process for calculating resisting forces: the *FS* and the tolerance (ep in Fig. 4), for which 1.2 and 1 were adopted, respectively. M_{α} was then calculated with (8) and substituted into (7) to obtain a new *FS*. This process continued until the error fell below the tolerance threshold or a maximum number of iterations, i.e., 40, was reached. If the maximum number of iterations is reached without converging, the corresponding slip circle was omitted.

Using this code, 12,720 slope images per category were created, 5/6 of which were used for training and validation, and the rest were used in testing. A total of 114,480 simulated slope images for slopes of different geometries and soil properties were created. One challenge in producing enough images for training and testing was that the ranges of *FS* values for the first and the last categories were much wider than the other seven categories as shown in Table 1. As a result, most of the produced images fell into these two categories, and images about three times that of needed were generated. To get categories with an equal number of samples, the redundant images were removed. In this process, the theory for the second type of error was utilized to improve the deep learning accuracy for the categories with fewer samples.

C. DATA OF SOIL PROPERTIES

Soil properties determine the stability of slopes and thus need to be included in the data. Therefore, the most influential parameters, which are also used in the SBM, were incorporated into the image data, i.e., the cohesion (c'), the friction angle (φ'), and the unit weight (γ). The cohesion is the molecular attraction between the soil particles; the friction angle is a measure of friction shear resistance of the soil. These two parameters demonstrate the shear strength of soil and have a great influence on its engineering behavior. Along with the normal effective stress, the strength parameters were used to describe the Mohr-Coulomb failure criterion representing the maximum shear resistance of soils. Additionally, unit weight was used to obtain the weight of soil and its resultant force in the slope. Equations (7) and (8) show the direct effect of these soil properties on determining the FS for each slope. The use of these three soil properties for slope stability analysis is standard practice in geotechnical engineering. Table 2 lists the ranges of these parameters, which were selected based on the typical values for each of them to reflect their common variations in typical applications [45], [46].

To obtain slopes with soil properties that are welldistributed in the ranges described above, random numbers were generated within these ranges for each soil property. The random values were then scaled and normalized to values between zero and one, in which zero and one represent the minimum and maximum values of a given property,

Soil properties	Range	Unit
Cohesion (c')	10-100	kPa
Friction angle (ϕ)	15-35	degrees
Unit weight (γ)	17-22	kN/m ³

TABLE 2.	Range of soil	properties	used in	the study.

TABLE 3.	Examples of soil	properties and their	associated RGB tri	plets
	Examples of son	properties and men	ussociated nob th	pices

Cases	Cohesion	Unit Weight	Friction Angle
	20 kPa	20 kN/m ³	32°
	R=0.11	G=0.6	B=0.85
	50 kPa	21 kN/m ³	25°
	R=0.44	G=0.8	B=0.5
	90 kPa	19 kN/m ³	19°
	R=0.88	G=0.4	B=0.2

respectively. Considering there are three soil properties, each of them was associated with one RGB channel of the image. As a result, each image can be viewed as three twodimensional arrays corresponding to the Red (R), Green (G), and Blue (B) channels (or components of each pixel). The number of elements in each array is identical to the number of pixels in the image, i.e., 227×227 . After linking cohesion with red, unit weight with green, and friction angle with blue, a color was created based on the randomly generated soil properties. In Table 3, three examples of slopes with typical soil properties are presented. The normalized soil properties as RGB pixel values are listed under the corresponding soil properties.

Real images of slopes carry information for soil properties in much more complicated ways, though there are correlations between the image pixels and the real soil properties. In this study, we adopted the above approach to simply demonstrate the concept and to test the feasibility of conducting deep learning with images containing soil property information. Despite the types and complexities of images, the ideas behind deep learning with artificial image data and real slope photos are the same: let CNNs extract the information directly from the images for classifications.

D. DATA OF GEOMETRIC PROPERTIES

The geometric properties of slopes are the other type of information influencing *FS* results for slopes. Similar to adding soil properties to computer code, random numbers were used to consider the effect of variations of geometry on the *FS*.



FIGURE 5. Example of slope image data.

As displayed in Fig. 5, slopes were generated in 50 m × 50 m rectangular regions (canvas). The crown and base of the slopes were assumed to be horizontal, hence $y_1 = y_2$ and $y_3 = y_4$. Also, the *x*-coordinates of Point 1 and Point 4 were fixed at 0 and 50, respectively (Table 4). Four random numbers, λ 's, were employed to generate the remaining coordinates of these four points. λ was a normalized variable between zero and one, which was generated with a uniform probability distribution.

Geometric information is also needed in the search for the most dangerous slip surfaces in the implementations of SBM. Three parameters of the slip circle, i.e., *x*-coordinate of the center, *y*-coordinate of the center, and its radius, affect the stability analysis. In this study, the *x*-coordinate of the center was assumed to be between 0 and x_3 , and the *y*-coordinate was assumed to be between y_3 and $y_3 + 15$.

As for the radius, the minimum value was the distance between the center and Point 3, while the maximum is the distance between the center and Point 1 or that between the center and Point 4, whichever is smaller. These upper and lower boundaries were then used to form a gridline with 30 grids in each direction. Consequently, 900 potential critical slip circle centers were evaluated for each slope. Iterations were also used to test 30 possible radii. The *FS* values were then calculated for these 27,000 critical slip circles, and the minimum value was chosen as the *FS* of the slope. It is noted that critical slip surfaces with a depth of less than 1 m, i.e., shallow failures, were neglected.

E. VALIDATION OF THE DATA

To assess the accuracy of the *FS* values obtained with the code, ten images per category were first randomly selected. The selected images were then analyzed using the SBM in a commercial program, i.e., Slide2, using the soil properties and geometries of the slopes. Identical SBM parameters, including α values, weights of slices, *FSs*, and critical slip circles, were adopted for the new code and Slide2. Figure 6 compares the *FS* values calculated by the two tools, in which the red line represents the *FS* values obtained by the newly developed code, and the blue circles represent the *FS* values

from RocScience. As can be seen, the results obtained with the new code agree well with those predicted by Slide2. Table 5 contains more details to highlight the accuracy of the code in nine random cases. Each row gives out detailed information for a slope belonging to one of the defined categories. As can be seen, the (relative) differences are about 0.1% - 0.3% in most cases.

F. PRETREATMENTS OF DATA

Slope image data was pretreated with several image preprocessing techniques to facilitate or/and improve deep learning with CNNs. These steps and their effects on the original image are demonstrated in Fig. 7. First, histogram equalization was applied to images to adjust the intensity of images and enhance their contrast. This technique better distributes intensities on the histogram and increases the local contrast of those areas with low contrast. Second, all slope images were resized to 227×227 pixels so that CNNs take input with the same size. These sizes were chosen to strike a balance between the learning outcome and computing demand. While it was not possible to show the original and resized size of the slope in Fig. 7, the proportions of the two images were kept to show the effect of this step. Third, images were stored in the format of the Lightning Memory-Mapped Database (LMDB), which is needed for Caffe. LMDB is a Btree based database management library, which is high-performance, memory efficient, and simple. The blue rectangle represents

TABLE 4.	Parameters	for g	generating	the	image	data	of slo	pes
								F



FIGURE 6. Comparison of *FSs* calculated with new computer code and RocScience.

this step in Fig. 7. Fourth, the mean image of training data was produced and subtracted from each input image for each feature to have a similar range. This could improve the learning outcome to some extent. An example showing the effect of these pretreatment steps is demonstrated in Fig. 7.

V. EXPERIMENTS AND RESULTS

A. DATA STATISTICS

Results for the generated data were assessed before discussing training and validation. The generated data in this

Parameter	Description	Formulation
(x_1, y_1)	Coordinates of Point 1	$(0,15+10\lambda)$
(x_2, y_2)	Coordinates of Point 2	$(15+10\lambda, y_1)$
(x_3, y_3)	Coordinates of Point 3	$(x_2 + \lambda(29 - x_2), y_2 + 8 + \lambda(35 - y_2))$
(x_4, y_4)	Coordinates of Point 4	$(50, y_3)$
N	Number of slices	40

TABLE 5.	Compared FS values o	f nine random cases	between RocScience a	nd our computer code.
INDER J.	compared i 5 values o	i mile fundom cuses	between noeselence a	na our compater coae.

Cartas	Cohesion	Unit	Friction	Dim	ensions of	Points 2 c	and 3	FS	FS
Category	(kPa)	(N/m ³)	angle (°)	<i>x</i> ₂	${\mathcal{Y}}_2$	<i>x</i> ₃	\mathcal{Y}_3	Code	Slide2
First	21	19	17	19.06	17.39	21.38	32.38	0.575	0.571
Second	31	22	25	23.83	15.57	28.02	31.20	0.828	0.827
Third	56	18	17	18.75	21.31	22.01	41.23	0.950	0.949
Fourth	16	17	25	16.70	22.62	28.94	37.80	1.031	1.030
Fifth	35	17	21	19.56	15.58	26.05	31.08	1.101	1.099
Sixth	55	18	21	15.84	19.01	28.85	41.92	1.218	1.219
Seventh	26	22	31	22.40	16.61	27.27	26.51	1.311	1.313
Eighth	67	17	32	17.81	15.43	27.01	39.61	1.463	1.464
Ninth	71	22	20	20.68	24.98	27.54	38.70	1.693	1.686



FIGURE 7. Pretreatment steps. Each sub-figure is the direct result from each step for computational purposes to demonstrate the original results in the pretreatment process.

TABLE 6. Data characteristics of all the data.

Category	, Coh (k	esion Pa)	Friction (degr	n angle rees)	Unit v (kN/	veight (m³)	Slo (deg	ppe rees)
	μ	σ	μ	σ	μ	σ	μ	σ
First	23.157	10.643	22.635	5.649	19.719	1.703	79.490	7.643
Second	33.623	13.758	24.386	5.945	19.657	1.689	77.425	8.960
Third	39.988	16.204	24.493	6.011	19.663	1.705	76.862	9.340
Fourth	46.643	18.204	24.658	5.964	19.613	1.711	76.775	9.559
Fifth	52.600	19.868	24.942	5.997	19.613	1.711	76.297	9.891
Sixth	58.050	20.470	25.244	5.943	19.574	1.704	76.050	10.018
Seventh	62.143	20.813	25.442	5.966	19.505	1.697	75.142	10.182
Eighth	65.432	20.971	25.801	5.963	19.448	1.704	74.581	10.316
Ninth	73.809	19.146	26.108	6.028	19.279	1.699	67.174	12.506
All	50.605	23.736	24.856	6.018	19.563	1.707	75.533	10.415

study is unique due to its close relevance to physics. From a physics perspective, the slope image data, which is labeled by the FS, is primarily dependent on two categories of parameters, i.e., material (soil) properties and geometric properties, as introduced in Section IV. In detail, three material property parameters (Table 2), i.e., cohesion, unit weight, and friction angle, and four (independent) geometric parameters (Table 4), i.e., four are independent parameters out of x_1 to x_4 and y_1 to y_4 , determine the labels of the image data, i.e., FS. Among them, the material property parameters have general physical meanings, while geometric parameters have specific meanings case by case, though this study selected a widely accepted setup in geotechnical engineering for 2D slope geometries. As introduced, all of these seven parameters were generated using a uniform distribution of probability from their individual ranges (Table 2 for material properties and Table 4 for geometric properties). Data generation according to the physics and procedure led to an imbalanced dataset: nine categories with different image numbers. In order to obtain balanced training and testing datasets, an unbalanced dataset with a total number of images that were three times that of the images to be used were generated. A random resampling was carried out to obtain a balanced dataset from the unbalanced one. As a result, it is worthwhile to assess the data statistics to understand the nature of the generated data. Because the training and testing datasets were generated in the same way and thus independent and identically distributed, the training, testing, and total datasets should have similar data statistics. Considering this fact, some major data statistics of all the data are presented in Table 6.

TABLE	7. Dee	p learning	g parameters.

Parameter	Value
Learning rate policy	fixed
Base learning rate	0.05
Test iteration	160
Test intervals	500
Momentum	0.9
Snapshot	5000
Weight decay	0.0001
Maximum iteration	40,000
Display	100
Solver mode	GPU

As can be seen, two common data characteristics, e.g., the mean and standard deviation, of the three general material property parameters and one general, derived geometric feature, i.e., slope angle, which was extracted from the four geometric parameters, were listed to give out the key data characteristics. Clear and distinct trends can be observed for different parameters. For example, the means of the cohesion and friction angle increase with the *FS* (category number), while the means of the unit weight and slope decrease with the *FS*.

B. TRAINING

The AdaDelta optimizer was adapted as the optimization method considering its low computation overhead compared to vanilla stochastic gradient descent and its successful application in the MNIST digit classification task dataset. Other advantages of AdaDelta include its low sensitivity to the hyperparameters, automatic learning rate configuration, use of separate dynamic learning rate per dimension, and robustness to large gradients, noise, and choice of architecture [47]. Hyperparameters with considerable influence on the learning process were initialized in addition to the optimizer. These hyperparameters are described in Table 7.



FIGURE 8. Variation of accuracy and loss in the training process.



FIGURE 9. Results of testing for the trained model.

Transfer learning was adopted as the training method, which can utilize knowledge embedded in a pre-trained CNN. To achieve the goal, weights from the BAIR Reference CaffeNet, which was a CNN pre-trained on the ImageNet dataset with 1000 classes, were used in the training of the CNN in this study. The BAIR Reference CaffeNet is slightly different from AlexNet [36]. Although the type of images in the ImageNet dataset is different from slope image data, transfer learning still helped reduce the training time and improve the accuracy significantly. This was very likely because both the ImageNet images and the slope images adopted in this study shared many basic features such as lines and color blocks. As shown in Fig. 8, training the CNN on 79,500 images and cross-validating it with 15,900 images yielded an accuracy of 82.74% and a loss of 0.3. This splitting ratio (5:1) was adopted to ensure comprehensive and sufficient training data while leaving ample data for cross-validation.

C. TESTING

After training, the model was tested on 19,080 slope images, which were independent of the training data, to evaluate the performance of the trained CNN. An accuracy of 79.45% and a mean absolute error of 0.2105 were achieved in the



FIGURE 10. Number of images versus the error in predicted FS.

testing. The result of this test is presented in Fig. 9. In this figure, the x-axis is the actual FS of the images that were used for testing, and the y-axis is the FS predicted by the trained model. The colored cells (squares) represent points with one or more images (or cases), while blank (white) cells indicate that there are no cases. Also, numbers are given in colored cells to show the number of images with those predictions. The green cells on the diagonal contain cases in which the predicted FS is identical to the actual FS. This green area corresponds to the accuracy of 79.45% obtained in the testing process. However, the deep learning accuracy value does not fully represent the accuracy of the slope stability analysis. This is because predictions that miss by 0.1 or more, i.e., cells slightly off the diagonal, could also be acceptable in practical applications, and such cases were not considered when calculating the accuracy of deep learning. The distance between green cells and other cells indicates the magnitude of the prediction error, while the colormap represents the number of cases in each cell.

Figure 10 is constructed to assess the performance of the trained CNN from a different perspective. This figure shows the distribution of the errors from a slope stability analysis perspective. A log scale was used for the y-axis to illustrate the error distribution. In 15,159 cases, the predicted and actual *FS* values are exactly the same in 79.45% of the testing data – that is how testing accuracy in deep learning was defined. Further analysis revealed that, in 3851 cases, i.e., 20.18% of the testing data, the predicted *FS* values are just one category (or 0.1) away from the actual one.

Figure 11 shows the results of the *FS* predictions using our proposed deep learning model for different categories of *FS* values. This figure compares the numbers of correct predictions with those that are off by one category and cases with more than one category. It is noteworthy that the total number of images in all categories is equal in the testing dataset, so the cumulative height of the three bars in each *FS* range is constant (Total number of images for each *FS* value range = 2120). This figure also shows that the predictions for the first and last categories are more accurate compared to those in the middle.



FIGURE 11. Results of FS predictions using the proposed deep learning model for different categories of FS values.



FIGURE 12. Computing time for the deep learning model and the LEM method.

D. PERFORMANCE EVALUATION AGAINST SBM

As this study is the first to use deep learning techniques for slope stability analysis, we thus validated our results using the traditional physics-based method. Specifically, we chose the most commonly used SBM, which is one of the LEMs that have been widely adopted to evaluate the *FS* of slopes due to their simplicity in estimating the landslide hazard. We compared the accuracy as well as the computation demand and time efficiency of our deep learning method to the traditional SBM.

For the deep learning testing data, the *FS* results were computed and validated using a traditional method, SBM, as discussed in Section IV-E. Thus, the testing results in Section V-B show the accuracy of the deep learning methods compared with those from the traditional method SBM. It is noteworthy to mention that the purpose of testing is to evaluate the performance of the trained model on unlabeled data. Therefore, the model input in the testing phase is new unlabeled image data to ensure consistent testing methodology.

In addition to accuracy, the computing demand and time efficiency of deep learning in comparison with the traditional methods also deserve great attention. Once the model is trained on a comprehensive dataset, deploying the model to calculate the FS values of new slope images is much faster than using conventional methods of slope stability analysis, e.g., LEMs. A comparison test was conducted on a workstation with a CPU of AMD Opteron 6386 SE Abu Dhabi 2.8GHz (64 Cores Total) and an internal memory of 128GB $(4 \times 32 \text{GB})$ DDR3 to assess the computing efficiencies of the traditional methods and the deep learning method. In Fig. 12, the computing time for the SBM is compared against that for the testing phase of the deep learning method. It should be noted that both models were tested using the CPU of the same computer. However, deep learning models can use the processing power of GPU, though, in this study, the GPU was an entry-level GPU (NVIDIA Quadro K600 2GB). In the analysis of a single image, although the deep learning method outperforms the conventional method, the difference in computing time is not significant; however, as the amount of data increases, the difference in the needed computing times gets more substantial. In the case of 200 images, the deep learning method is more than 90 times faster than the traditional method.

VI. DISCUSSION

In this study, deep learning methods were adopted to explore their potential as an alternative approach to slope stability analysis. Deep learning methods are an inevitable and enticing tool in the future of engineering. These models can benefit from the era of big data and new sensor networks and technologies to provide a complimentary, if not alternative, tool for future engineers. The proposed method addresses some of the issues associated with the traditional methods, including the adoption of assumptions and limitations in considering complex real-world problems.

The performance of the new method was analyzed in terms of accuracy and computational efficiency. It was found that the accuracy of the new method is 79.45% in the deep learning perspective. Further analysis showed that the majority of incorrect predictions belong to those that are slightly off the correct prediction. That is, from the 20.55% of incorrect predictions, 20.18% belong to those that are off by 0.1 in terms of FS. This is noteworthy because the process of transforming FS (continuous variable) to FS ranges/classes (discrete variable) leads to the loss of information. That is, the deep learning model is unable to distinguish between the FS values of images that belong to the same range/class. It gets further complicated when the model handles images with FS values that are close but belong to two nearby classes/ranges. Therefore, if predictions with just one category apart are considered to be correct, the accuracy of testing will increase to 99.63%. This accuracy is satisfactory, considering that the difference between FS predictions made with different LEMs and strength reduction methods usually vary considerably in a 5% range [12]. This high accuracy of the new deep learning method in slope stability analysis provides a compelling

reason to further investigate deep learning as a promising future direction for traditional stability analysis.

Additionally, the higher accuracy of the categories on the two ends versus those on the middle in Fig. 11 can be explained using the theory presented in Section III. To facilitate explanation, this observation can be first formulated mathematically. That is, for an "end" category i, $p(B|A_i) >$ p(B) or $\frac{p(B|A_i)}{p(B)} > 1$. According to (6), we just need to show $\frac{p(A_i|B)}{p(A_i)} > 1. p(A_i|B)$ and $p(A_i)$ are the probabilities that this category of images appears in the training data (balanced) and that in all the available data (unbalanced), respectively. As explained in Section V-A, fewer images with very high or low FS values, i.e. categories on the two ends, tend to be generated (i.e., in all the available data) based on the underlying physics and given data generation procedure. As a result, the probability of an image sample belonging to an "end" category in all the available data, $p(A_i)$, is lower than that in the training data obtained via resampling, $p(A_i|B)$. That is, $\frac{p(A_i|B)}{p(A_i)} > 1$. This proves the theory that was proposed in this study can be employed in future studies to intentionally improve the accuracy of interested FS ranges.

In addition to accuracy, one of the main advantages of CNNs is their capability to analyze large amounts of data within a relatively short amount of time. Figure 12 in the results section offers an insight into the high potential and efficiency of the new method. This difference would be much more significant if the amount of time needed for the user to construct a LEM model and prepare input parameters is included. By contrast, the deep learning model can be deployed and used on the raw data without manual model construction or treatments. For example, for each case (corresponding to an image), LEMs need time to set up geometry, and SRMs require lots of effort to prepare the models for numerical analysis. The above comparison demonstrated the computational efficiency of the proposed deep learning method compared with traditional LEMs in addition to its flexibility and robustness in dealing with raw image data.

It is also important to mention that this study was intended as a proof of concept and thus adopts 2D cross-sections of simulated data in a well-controlled setting as training and testing samples. Therefore, a lot must be done before it is ready for commercial use and real-world implementations. However, the current results suggest that deep learnings can perform as well as physics-based methods if they are provided with enough data.

VII. CONCLUSION

In this study, we developed an image-data-driven method for slope stability analysis for landslide prevention, which is the first to use deep learning to predict FS of slopes that is a key parameter for estimating landslide occurrences. The proposed method is capable of predicting FS for a given slope while considering its soil property information as well as geometry and critical slip circle from slope images. The results showed that the prediction accuracy could reach 99.63%, and the

computing time was considerably less than that of the traditional SBM. It demonstrates the high feasibility of achieving fast in-situ decision-making for natural hazard analysis and prevention with the aid of the internet of things and big data analytics.

REFERENCES

- [1] D. J. Varnes, "Landslide types and processes," Landslides Eng. Pract., vol. 24, no. 29, pp. 20–47, 1958.
- [2] R. L. Schuster and M. Lynn, "Socioeconomic and environmental impacts of landslides in the western hemisphere," in *Proc. 3rd Panamer. Symp. Landslides*, vol. 29, 2001, pp. 1–50.
- [3] D. P. Kanungo, M. K. Arora, S. Sarkar, and R. P. Gupta, "A comparative study of conventional, ANN black box, fuzzy and combined neural and fuzzy weighting procedures for landslide susceptibility zonation in Darjeeling Himalayas," *Eng. Geol.*, vol. 85, nos. 3–4, pp. 347–366, Jun. 2006.
- [4] M. Ercanoglu and C. Gokceoglu, "Use of fuzzy relations to produce landslide susceptibility map of a landslide prone area (West Black Sea Region, Turkey)," *Eng. Geol.*, vol. 75, nos. 3–4, pp. 229–250, 2004.
- [5] A. W. Bishop and N. Morgenstern, "Stability coefficients for earth slopes," *Geotechnique*, vol. 10, no. 4, pp. 129–153, Dec. 1960.
- [6] O. C. Zienkiewicz, C. Humpheson, and R. W. Lewis, "Associated and non-associated visco-plasticity and plasticity in soil mechanics," *Geotechnique*, vol. 25, no. 4, pp. 671–689, Dec. 1975.
- [7] A. Pourkhosravani and B. Kalantari, "A review of current methods for slope stability evaluation," *Electron. J. Geotech. Eng.*, vol. 16, pp. 1245–1254, Jan. 2011.
- [8] W. Fellenius, "Calculation of stability of earth dam," in *Proc. 2nd Congr. Large Dams*, Washington, DC, USA, vol. 4, 1936, pp. 445–462.
- [9] A. W. Bishop, "The use of the slip circle in the stability analysis of slopes," *Geotechnique*, vol. 5, no. 1, pp. 7–17, Mar. 1955.
- [10] N. R. Morgenstern and V. E. Price, "The analysis of the stability of general slip surfaces," *Geotechnique*, vol. 15, no. 1, pp. 79–93, Mar. 1965.
- [11] J. M. Duncan and S. G. Wright, "The accuracy of equilibrium methods of slope stability analysis," *Eng. Geol.*, vol. 16, nos. 1–2, pp. 5–17, Jul. 1980.
- [12] D. Resendiz, "Accuracy of equilibrium slope stability analysis," Int. J. Rock Mech. Mining Sci. Geomech. Abstr., vol. 11, no. 11, p. A231, Nov. 1974.
- [13] Y. M. Cheng, T. Lansivaara, and W. B. Wei, "Two-dimensional slope stability analysis by limit equilibrium and strength reduction methods," *Comput. Geotechnics*, vol. 34, no. 3, pp. 137–150, May 2007.
- [14] S. Oh and N. Lu, "Slope stability analysis under unsaturated conditions: Case studies of rainfall-induced failure of cut slopes," *Eng. Geol.*, vol. 184, pp. 96–103, Jan. 2015.
- [15] A. H. Alavi and A. H. Gandomi, "A robust data mining approach for formulation of geotechnical engineering systems," *Eng. Comput.*, vol. 28, no. 3, pp. 242–274, Apr. 2011.
- [16] I. A. Basheer, "Selection of methodology for neural network modeling of constitutive hystereses behavior of soils," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 15, no. 6, pp. 445–463, Nov. 2000.
- [17] H. Park, "Study for application of artificial neural networks in geotechnical problems," in *Artificial Neural Networks-Application*. Norderstedt, Germany: Books on Demand, 2011.
- [18] M. Huang and C.-Q. Jia, "Strength reduction FEM in stability analysis of soil slopes subjected to transient unsaturated seepage," *Comput. Geotechnics*, vol. 36, nos. 1–2, pp. 93–101, Jan. 2009.
- [19] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," Int. J. Inf. Manage., vol. 35, no. 2, pp. 137–144, 2015.
- [20] F. Shaheen, B. Verma, and M. Asafuddoula, "Impact of automatic feature extraction in deep learning architecture," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov. 2016, pp. 1–8.
- [21] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, pp. 5455–5516, Apr. 2020.
- [22] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Inf. Fusion*, vol. 42, pp. 146–157, Jul. 2018.
- [23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [24] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," J. Big Data, vol. 2, no. 1, pp. 1–12, Dec. 2015.

- [25] F. Mayoraz and L. Vulliet, "Neural networks for slope movement prediction," *Int. J. Geomech.*, vol. 2, no. 2, pp. 153–173, 2002.
 [26] F. T. de Souza and N. F. F. Ebecken, "A data based model to predict
- [26] F. T. de Souza and N. F. F. Ebecken, "A data based model to predict landslide induced by rainfall in Rio de Janeiro city," *Geotech. Geol. Eng.*, vol. 30, no. 1, pp. 85–94, Feb. 2012.
- [27] D. Kawabata and J. Bandibas, "Landslide susceptibility mapping using geological data, a DEM from ASTER images and an artificial neural network (ANN)," *Geomorphology*, vol. 113, nos. 1–2, pp. 97–109, Dec. 2009.
- [28] I. Flood and N. Kartan, "Neural networks in civil engineering. I: Principles and understanding," J. Comput. Civil Eng., vol. 8, no. 2, pp. 131–148, Apr. 1994.
- [29] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, arXiv:1511.08458.
- [30] E. J. Lee, S. Y. Shin, B. C. Ko, and C. Chang, "Early sinkhole detection using a drone-based thermal camera and image processing," *Infr. Phys. Technol.*, vol. 78, pp. 223–232, Sep. 2016.
- [31] K. Makantasis, E. Protopapadakis, A. Doulamis, N. Doulamis, and C. Loupos, "Deep convolutional neural networks for efficient vision based tunnel inspection," in *Proc. IEEE Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2015, pp. 335–342.
- [32] Z. Liu, S. Hu, Y. Sun, and B. Azmoon, "An exploratory investigation into image-data-driven deep learning for stability analysis of geosystems," *Geotech. Geol. Eng.*, vol. 39, pp. 1–16, Aug. 2021.
- [33] D. V. Dao, A. Jaafari, M. Bayat, D. Mafi-Gholami, C. Qi, H. Moayedi, T. V. Phong, H.-B. Ly, T.-T. Le, P. T. Trinh, C. Luu, N. K. Quoc, B. N. Thanh, and B. T. Pham, "A spatially explicit deep learning neural network model for the prediction of landslide susceptibility," *Catena*, vol. 188, May 2020, Art. no. 104451.
- [34] Z. Karim, R. Hadji, and Y. Hamed, "GIS-based approaches for the landslide susceptibility prediction in Setif region (NE Algeria)," *Geotech. Geol. Eng.*, vol. 37, no. 1, pp. 359–374, Jan. 2019.
- [35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 1998, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

- [37] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
- [38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [39] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACMMM*, 2014, pp. 675–678.
- [40] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Comput.*, vol. 22, no. 12, pp. 3207–3220, 2010.
- [41] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017.
- [42] K. Chellapilla, S. Puri, and P. Simard, "High performance convolutional neural networks for document processing," in *Proc. 10th Int. Workshop Frontiers Handwriting Recognit.*, 2006, pp. 1–7.
- [43] B. S. Firincioglu and M. Ercanoglu, "Insights and perspectives into the limit equilibrium method from 2D and 3D analyses," *Eng. Geol.*, vol. 281, Feb. 2021, Art. no. 105968.
- [44] R. L. Michalowski, "Limit analysis and stability charts for 3D slope failures," J. Geotech. Geoenviron. Eng., vol. 136, no. 4, pp. 583–593, Apr. 2010.
- [45] R. B. Peck, W. E. Hanson, and T. H. Thornburn, *Foundation Engineering*. Hoboken, NJ, USA: Wiley, 1974.
- [46] Characteristic Coefficients of Soils, Association of Swiss Road and Traffic Engineers, Standard 670 010b, Swiss Association for Standardization, 1999.
- [47] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, arXiv:1212.5701.
- [48] J. Sun and Z. Zhao, "Stability charts for homogenous soil slopes," J. Geotech. Geoenviron. Eng., vol. 139, no. 12, pp. 2212–2218, Dec. 2013.
- [49] STB2010. A Software for Slope Stability Analysis. Accessed: Feb. 2021. [Online]. Available: https://www.geoengineer.org/software/stb

• • •