IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 12, No. 3, September 2023, pp. 1428~1438
ISSN: 2252-8938, DOI: 10.11591/ijai.v12.i3.pp1428-1438

Deep learning based slope erosion detection

Caio Henrique Esquina Limão[1], Thabatta Moreira Alves de Araújo[2], Carlos Renato Lisboa Francês[1]
[1]Institute of Technology–ITEC, Federal University of Pará, Belém, Pará, Brazil
[2]Department of Informatics, Management and Design–DIGD, Federal Center for Technological Education of Minas Gerais - CEFET-MG, Belo Horizonte, Minas Gerais, Brazil

Article Info

Article history:

Received Jul 4, 2022
Revised Sep 13, 2022
Accepted Oct 25, 2022

Keywords:

Artificial intelligence
Convolutional neural network
Deep learning
Erosion
Image classification
Slope
Structure health monitoring

ABSTRACT

Being increasingly present at the most diverse structure health monitoring (SHM) scenarios, many high-performance artificial intelligence techniques have been able to solve structural analysis problems. When it comes to image classification solutions, convolutional neural networks (CNNs) deliver the best results. This scenario encourages us to explore machine learning techniques, such as computer vision, and merge it with different technologies to achieve the best performance. This paper proposes a custom CNN architecture trained with slope erosion images that showed satisfactory results with an accuracy of 96.67%, enabling a precise and improved identification of instability indicators. These instabilities, when detected in advance, prevent disasters and enable proper maintenance to be carried out, given that its integrity directly affects structures built around and above it.

Corresponding Author:

Caio Henrique Esquina Limão
Institute of Technology – ITEC, Federal University of Pará
Curuzu street, Mares do Sul building, Zip Code: 66085-823. Pará, Brazil
Email: caiolimao97@gmail.com

1. INTRODUCTION

Slope failures can cause environmental, economic, psychological, property damage and even result in the loss of human life. That is why in most civil engineering projects, site stability has been considered a major problem. Unfortunately, even the soil erosion being one of the major causes of this kind of instability, it's often neglected, showing it results in the long-term [1].

Soil erosion, by definition, is a geomorphic and, at the same time, a land degradation process that detaches soil particles, rock fragments, soil aggregates and organic matter from its primary location and then transports these to another location. Being affected, and many times intensified, by human intervention, the natural erosion processes have presented relevant increases in their rates of occurrence across all kinds of landscapes [2], [3]. This kind of land degradation is the major cause of slope instability and it occurs when the soil resistance drops. This decrease is mainly caused by: reduction of matrix suction stress, discontinuities (faults, joints and fractures), modification of the structure of sensitive soils, liquefaction of fine saturated sand and loss of cohesion [4].

This paper provides a computer vision approach for slope structural damage detection, as it seeks to interpret and understand the visual world. The apparent damage on slopes is the most relevant object of this study, as erosions, discontinuities, animal dens and anthills are threats to its structural integrity. Since the damages caused to slope structures are, at some point, apparent, image analysis and anomaly detection through convolutional neural networks (CNNs) arises as a great option for damage identification, given that such occurrences are multifactorial, some of the failure modes are detected only visually, which justifies the use of computer vision techniques on those scenarios.

To address all the aforementioned issues, this article proposes a 3-layer CNN with 2 fully-connected (FC) layers to detect apparent structural damage on slopes and classify if it is stable, given the visual diagnosis of it. This kind of neural network enables the early detection of slippage and erosion occurrences, preventing accidents, maintaining the integrity of the slope and the environment in which it is found. A CNN-based image analysis and classification software was preferable due to the ease of merging it with existing monitoring systems such as drones, satellites and any other type of system that generates images, enhancing its precision, objectivity and processing speed [5].

The following sections are organized as: section 2 presents the related works alongside with the concepts of the main technologies and their role for the optimal functioning of the proposed approach, addressing their pros and cons as well as their current state. Section 3 describes the proposed network and how it works. Section 4 performance analysis of the developed CNN, highlighting the training and test details that presented the best result for the detection of slope instability. Finally, section 5 concludes the paper with the analysis results and improvements that this proposal needs to undergo.

2. RELATED WORKS AND BACKGROUND CONCEPTS

This section will present a brief literature review, including several researches and proposals about image detection artificial neural networks (ANNs) as well as slope stability prediction methods, as described on Table 1. For the development of this paper, the bibliographic research revolved around well-founded articles that would serve as the basis for new ANN implementations. Obviously, this paper seeks to merge the main themes and technologies of each and every article read to propose the most suitable architecture so that the best performance is achieved by the neural network.

Among the studied proposal, the following stand out: Yang *et al.* [6] contributed with an innovative proposal, training and testing deep learning (DL) models using the you only look once (YOLO) approach to analyze the synthetic data obtained from a graphical model of bolt loosening marks at wind turbine towers and achieving an accuracy of 95.71% with a detection time of 0.024 seconds for a single image using the YOLO v3. Liu *et al.* [7] presented an innovative method to improve the analysis process, denominated finite element (FELEM), slope stability analysis using elastic finite elements that can be analyzed by DL methods, such as CNN, but requires an extensive and reliable dataset to enable this approach.

Table 1. Used base related works

Studied research	Covered technologies					Research summary
	Slope stability detection	Deep learning	CNN	ReLU	Batch normalization	
Yang *et al.* [6]	X	√	√	X	√	Innovative proposal using YOLO v3 to achieve a fast and precise neural network for bolt loosening detection for wind turbine towers.
Liu *et al*. [7]	√	X	X	X	X	A two-dimensional (2D) and 3D slope stability FELEM using elastic finite element stress fields was proposed.
Qi and Tang [8]	√	√	X	X	√	Proposed an excellent comparison between six different machine learning (ML) algorithms.
Lu *et al.* [9]	X	√	√	√	X	Tested different types of CNN architectures and datasets to find the optimal plant leaf disease classification method.
Kattenborn *et al.* [10]	X	√	√	√	√	Well explained neural network model for vegetation classification.
Yadav and Jadhav [11]	X	√	√	X	√	Disease diagnosis using a CNN based architecture using different parameters and techniques to acquire the best performance.

Qi and Tang [8] proposed an amazing comparison between six ML algorithms, including logistic regression (LR), decision tree (DT), random forest (RF), gradient boosting machine (GBM), support vector machine (SVM), and multilayer perceptron neural network (MLPNN). Lu *et al.* [9], highlights the best scenarios for a CNN to be able to perform in a satisfactory way when it comes to image classification. And works like the one presented by Kattenborn *et al.* [10], a review on CNN in vegetation remote sensing, complements it. So that ideas like the one in this paper emerge, since these types of applications can arise in the most diverse areas.

Similar CNN solutions [12]–[18] differ from the one proposed in this paper given that the developed architecture, combined with the technologies presented in the following section, had better results and achieved greater performance when compared to the most common ones in the current literature. This is due to the need of multiple convolutional layers to properly process the available slope dataset and enable the system to be easily combined with already existing monitoring systems, using the images obtained by them (drones and satellites) to extract information regarding the visual integrity of the slope.

This work seeks to explore use of computer vision in conjunction with high-performance processing techniques, such as CNN, to detect damage in geotechnical structures, since it is essential to the reliability of these kinds of structures and others that depend on them. The next subtopics will provide the definitions and essential concepts needed for the solution proposed by this paper to be fully understood. They will be ordered: slope stability, DL for image classification, CNN particularities, activation function rectified linear unit (ReLU), Adam optimizer and batch normalization.

2.1. Slope stability

Slope stability assessment is crucial for geotechnics and civil engineering, as artificial slopes and the design of dams, embankments, and similar structures directly impact the ecosystem. Slopes are terrains that serve as a support base for the soil, and may have a natural or artificial origin (man-made). Non-natural slopes consist mostly of embankments built to prevent landslides in places where their stability is not efficient [19]. The topography elevation, aspect and angle of a slope are the most important factors in geomorphology, which directly influences slope stability. That being, the risk of landslides is higher in more steep slopes, where it is advised not to exceed 45º, since greater variations cannot guarantee its stability [20].

In order to resist the pressure generated by the earth, this type of containment is chosen, as it improves the soil through the execution of tie rods, anchors, shotcrete and drainage. There are several techniques that aim to protect a slope, and the choice of which one to use depends on the type of project designed. This protection can be made of stone coating, concrete, retaining wall, slope berm, anchorage, among other measures. Depending on the length of the slope, it is advisable to build contour lines to avoid erosion caused by rain. Another method of preservation is the use of vegetation to cover the slopes, providing greater stability [20], [21].

In this paper, slope instability is considered as the removal of the vegetation cover, soils and underlying loose material. An instability can be classified into two categories: landslides and erosion. The first one represents the gravitational mass movements that occur when the stress exceeds the mechanical resistance of a slope, and the second concept includes the removal of the vegetation and/or topsoil caused by different types of erosion [22].

2.2. Deep learning for image classification

DL is a concept that derives from the conventional neural network, but outperforms it by employing transformations and graph technologies simultaneously, becoming a multi-layer network. It outperforms other types of ML architectures and is able to process audio, images and natural language, among others [23]. Between all types of deep neural network models, when it comes to image processing, three stand out. These are: deep belief networks (DBNs), stacked autoencoders (SAEs) and CNNs [24].

For this proposal, a custom CNN based architecture is created because of the advantages that this type of neural network adds when it comes to image classification. Their structure was inspired by the actual operation of the vision itself, and has become a successful tool in computer vision and state-of-art models of neural activity and visual tasks. They start their process by convolving a set of filters with the input and rectifying the outputs, leading to “feature maps”, akin to the planes of S-cells in the neocognitron [25].

Basically, the convolution layer is responsible for convolving the image patches. Then, the pooling layer resizes the feature maps that resulted from the previous process, to get more abstract and universal features. And for the last step, these maps are transformed into vectors by the fully connected layer [26]. A representative structure of CNNs is shown in Figure 1, containing two convolutional-pooling layers and a fully-connected one.

2.3. CNN particularities

To decrease the amount of weight connections, and optimize the training/test step, a more efficient method emerged, it proposed to look for local regions on images instead of looking at each pixel. So, the hidden neurons in the next layer only get inputs from the corresponding part of the previous layer. For example, it can only be connected to 5×5 neurons. Thus, if we have 64×64 neurons in the next layer, then it will become 5×5×3 by 64×64 connections, which is 43,200 (instead of 50.331.648 to fully connect it). To further simplify the neural network connections, we can set the weights fixed for all neurons in the next layer, connecting neighboring neurons with the same weight they had for the region analyzed in the previous layer. Therefore, the parameters would undergo another significant reduction, resulting in only 5×5×3=75 to connect 64×64×3 neurons to 64×64 in the next layer (from 50.331.648 to 75 connections) [27]–[29].

As represented in Figure 2, the convolutional layer uses filters that perform convolution operations as it is scanning the input. These operations are performed on the entire image, by sliding and finding the dot product between the filter and the input image parts. The output of this operation is termed Feature Map, which provides the corners and edges information about the image and is read by other layers, so they can learn the remaining features of the image [29].

Commonly, images have a series of redundant information, therefore, the need to use the pooling layers inserted in several of the convolutional layers of the network, to avoid substantial performance degradation [30]. The best pooling technique to work with representations that rely on count statistics, such as bag-of-visual-words (BOV) ones has shown to be the one called max-pooling, because it reduces the size of the hidden layers by an integer multiplicative factor, improving performance in applications that process many images [31]–[33]. Max-pooling creates position invariance over larger local regions and down-samples the input image by a factor of K_x and K_y along each direction. Leading to a faster convergence rate by selecting superior invariant features which improves generalization performance [34], [35].

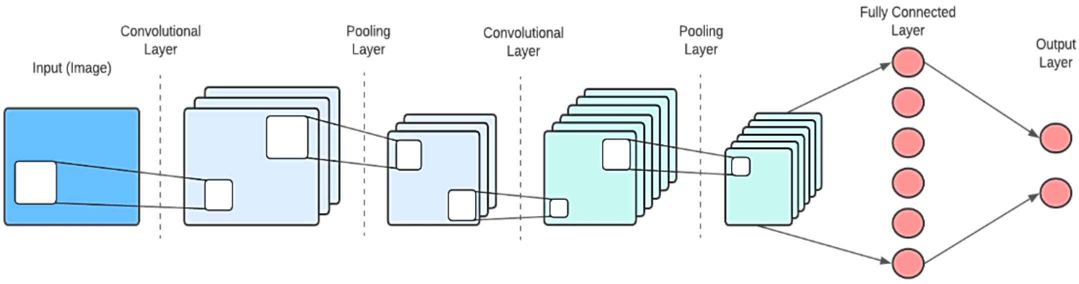

Figure 1. Illustrative CNN structure with two convolutional-pooling layers and one fully connected

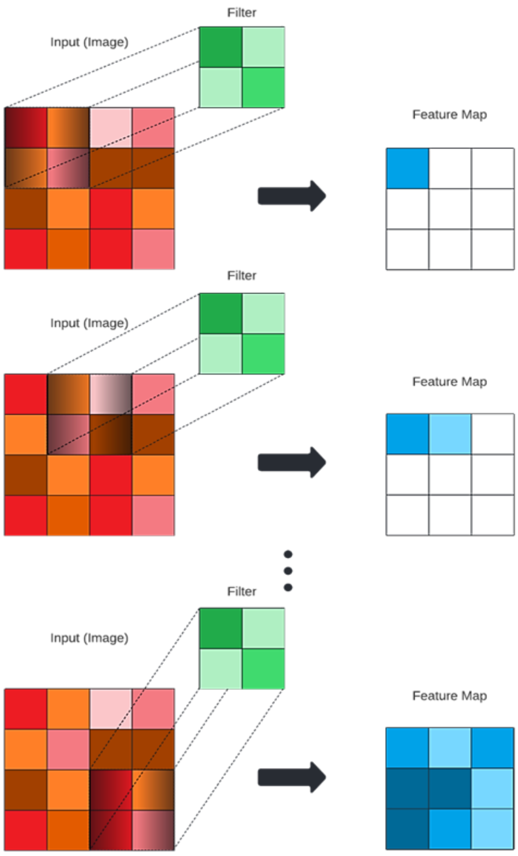

Figure 2. Convolutional layer flowchart

As for the last major particularity, we have the FC layer. It is known that the features generated by the last convolutional layer corresponds to a portion of the input, so, the network's perspective view does not cover the entire spatial dimension of the image. This lack of coverage makes the FC layer a mandatory part of the network. As for its parameters, such as the number of FC layers and number of neurons required in FC layers varies for each scenario [36].

2.4. Activation layer ReLU

Activation functions are commonly used after each convolution layer in order to achieve a certain linearity at the neuron's output, since image data are not linearly separable [37]. ReLU is the most used activation function in the world, both for CNNs and DL, it is the most efficient. As much as it appears to be a linear function, it is not, as it has a derivative function that allows backpropagation. However, when the input approaches zero or less, the gradient of the function becomes zero, preventing the network from performing backpropagation [38].

In practice, the outputs of the filters are submitted to the activation function at the end of each and every convolutional layer, and after going through this process, used to update the neural network weights [39]. By observing the mathematical expression of the ReLU function, $f(x) = \max(0, x)$, it can be seen that the neurons will only activate if the input is greater than zero, so, neurons that receive negative values will be "erased". This particularity increases the training speed and reduces the computational cost, however, if a neuron receives only negative values, it will not learn anything.

2.5. Adam optimizer

An adaptative optimization algorithm exists to find the best weights for a neural network, aiming for the minimization of the error function (the closest do zero, the best), reflecting in the reduction of the error of the network as a whole. The adaptive moment estimation (ADAM) combines the best properties of the AdaGrad and root mean square propagation (RMSProp) algorithms to provide an optimization algorithm that handles noisy problems [40]. It was invented by Kingma and Ba [41] and is one of the most popular step size methods in the area of neural networks. It converges much faster for multi-layer neural networks or CNNs, than any other optimizer, but it is not quite as good for generalization [42].

2.6. Batch normalization

The batch normalization (BN) allows the hyper parameters to be more freely defined, as it significantly reduces training time by normalizing the input of each layer in the network, and not only the input layer itself. This approach allows the use of higher learning rates, reducing the number of training steps [43]. These advantages make batch normalization a natural candidate to speed up training of different combinations of hyperparameters needed to optimize the use of dropout layers, as it makes the network converge faster [43]. During training, BN estimates the mean and variance of the entire activations within a mini-batch through exponential moving average with update factor, and in the testing phase, it uses those values for whitening input activations [44].

3. PROPOSED APPROACH

This paper proposes a 3-layer CNN architecture that detects different types of erosion and landslides, adding redundancy to the inspection process and often saving the displacement of professionals. On the other hand, it does not provide the calculation of the factor of safety, meaning that the slope stability assessment can't be proceeded by this method. It was not possible to perform this calculation due to the unavailability of a robust dataset that ensures data labeling for all the needed scenarios.

The dataset analyzed in this paper consists of a total of 300 images, 200 of them are for erosion images and 100 for slope structures without any visible damage, considered stable (without visible erosion or landslide). The distribution of the images has the following proportion: 73% for training (219 images), 17% for validation (51 images) and 10% for testing (30 images). The validation dataset differs from the test dataset because it is used to give an unbiased estimate of the skill of the final tuned model while tunning the hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration [45].

Figures 3 and Figure 4 represent the types of photos used by the network to learn how to identify an erosion and a stable slope. These illustrations are shown Figure 3 and Figure 4 just to clarify what is meant by "Erosion" and "Stable Slope" mentioned in this paperfor. The first term generalizes apparent damage that compromises or may come to compromise the slope structure, while the second one includes all kinds of minor occurrences that are considered irrelevant (like the holes dug for planting at Figure 4).

Figure 3. Example of an image for erosion [46]

Figure 4. Example of an image for a stable slope [47]

As shown in Figure 5, this paper proposes a custom CNN with 3 convolutional layers, The first and second ones have 32 filters of size 3 (height and width), each one of them having their output submitted to a 2×2 max pooling layer. As for the output of the third, and last, convolutional layer, it is noticed an addition of the flatten layer to the max pooling one (present on the first two convolutional layers). This flatten layer is responsible for reshaping a 4-dimension output into 2D, so the fully connected layers can utilize their neurons alongside with the ReLU function to extract useful information.

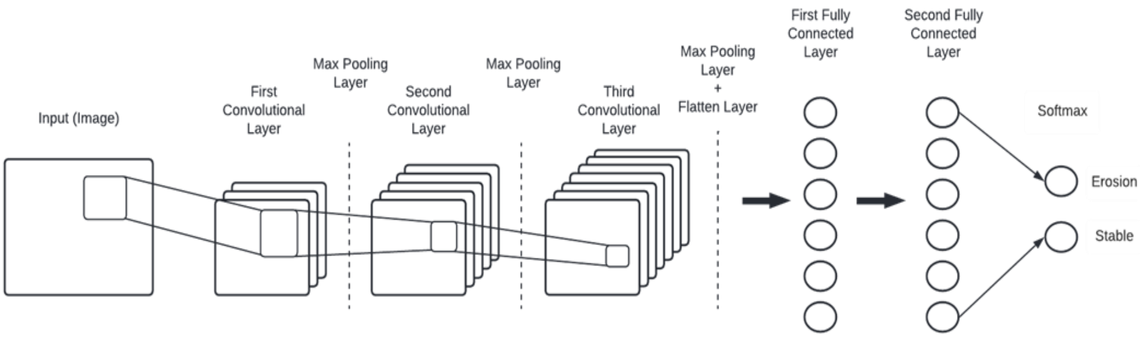

Figure 5. Custom CNN structure that got the best performance

For this application, it would not be interesting for the winning class to be defined by the highest numerical value. The output of the second fully connected layers uses the Softmax function to estimate the probability that each input has to belong to each of the two possible output classes (erosion and stable) with an interval between 0 and 1. Consequently, for each input submitted to the neural network, it returns the probability of having an instability on it, as well as the probability that the structure being stable.

4. PERFORMANCE ANALYSIS

Since the neural network flowchart developed for this application was explained in previous sections, this one brings more specifics details of the CNN used to solve the previously exposed issue. Among these details, there are: hyperparameters, convolutional weights, confusion matrix and overall performance, as some of them are shown in Table 2. To achieve the results discussed in this paper, the following environment setup was used: i) hardware: Windows 10 Home x64, 8 GB RAM and Intel i7-8750 H (2.21GHz); ii) programming language: Python 3.4.12; iii) web-based interactive computing platform: Project Jupyter; and iv) Main Python Libraries: TensorFlow 2.8.0, Pandas 1.4.1, NumPy 1.22.3, Matplotlib 3.5.1 and Scikit-learn 1.0.2.

Looking at Table 2, we can see that, in fact, the architecture proposed in this paper achieved the best performance. It is worth noting that each combined architecture and its respective hyperparameters were tested at least four times. This quantity of tests was carried out so that the random factor of the weights, and training, did not considerably affect the overview of each of the architectures.

Table 2. Comparison between different architectures

Architecture	Learning rate	Epochs/ iterations	Number of neuron son the FC layers	Batch size	Performance for the validation dataset	Performance for the test dataset
3 convolutional layers + max pooling after each of them + 2 FC layers	1e-4	96/7000	128	6	86.3%	29/30
4 convolutional layers + max pooling after each of them + 1 FC layers	1e-4	96/7000	128	6	84.31%	25/30
3 convolutional layers + max pooling after each of them + 2 FC layers	3e-4	96/7000	128	6	82.35%	25/30
3 convolutional layers + average pooling after each of them + 2 FC layers	0.001	96/7000	128	6	80.39%	24/30
2 convolutional layers + average pooling after each of them + 1 FC layer	0.01	96/7000	64	6	70.58%	18/30
2 convolutional layers + max pooling after each of them + 1 FC layer	0.001	96/7000	64	6	66.66%	18/30

The batch size was set to 6 due for a better split of the image samples from the slope dataset as well as the decrease of the computational cost. The use of 128 neurons in the architecture tests is preferable due to the number of pixels each image had after the reshape (128×128), which is justified, since in cases where 64 neurons were used, the performance dropped considerably. It is worth mentioning that all the tests used Adam optimizer, as it has been the best optimizer option for deep learning scenarios like the one proposed in this paper. As expected, the CNN success rate for the validation dataset and the test dataset are similar in most cases, even though both these datasets had their samples randomly separated from the 300 original images. To better understand the weights and outputs of each convolution layer, the original inputs are needed, those being the images analyzed by the proposed neural network. Figure 6 is an example of an input (landslide) that was already resized to 128×128 pixels and then analyzed by the CNN. Figures 7 and Figure 8 are the interpretation of the neural network for the submitted example in Figure 6, respectively representing the weights and filters as well as the feature maps that were generated by the first convolutional layer.

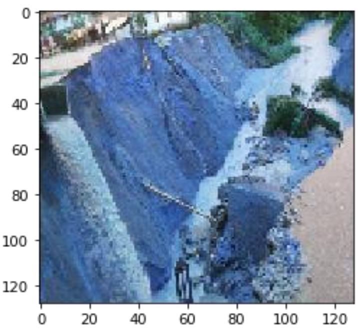

Figure 6. Example of an 128×128 image from the test dataset [48]

The examination of Figure 7 shows that the first convolutional layer traces the most relevant sections of the given example in Figure 6 according to the relevance presented by the filters during the mapping process. The red values have positive values and contribute to the interpretation of the input. On the other hand, the blue ones are negative, which means that the corresponding area of the image has a low relevance level to the definition of the final answer. As previously mentioned, 32 3×3 filters in Figure 8 were applied during the convolutional operations of the network. This filtering results in 32 different activation maps, or neuron matrices (the output itself). The analysis of the feature maps generated on each convolutional layer, seeks to understand which features the CNN detects, in this case, highlights with shading. For the last convolutional layer, the network achieves the 64 3×3 filters, predefined to enable the CNN to map each image in a more embracing way. As it is noticed, for the same input, the filters and feature maps of the third, and last, convolutional layer in Figures 9 and Figure 10 are more accurate, since they have already undergone the adjustments of all the previous convolutions and pooling operations. Therefore, the shadow effects better distinguish the limits of the slope and its instabilities.

Using the parameters exposed in this section, clarifies, through the confusion matrix, that the network was able to correctly identify 32 images that presented instability, of the 37 available on the validation dataset, and 12 out of 14 stable slopes, as shown Figure 11. As for the test dataset, 29 out of 30 images were precisely classified, where a third of them were representations of stable slopes and the rest of the images were split between erosion and landslide occurrences on a slope structure.

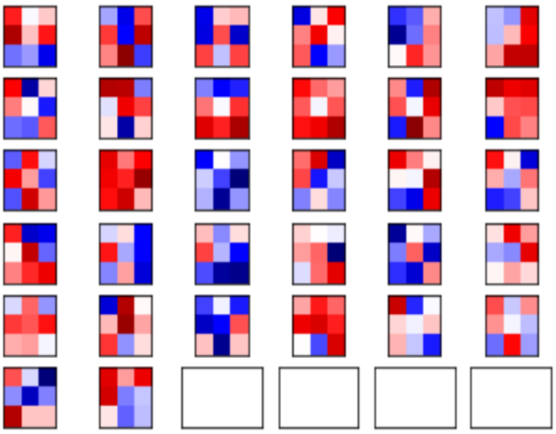

Figure 7. Filters of the first convolutional layer

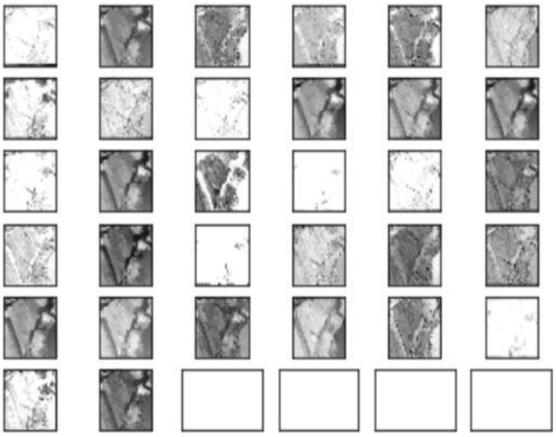

Figure 8. Feature maps generated by the first convolutional layer

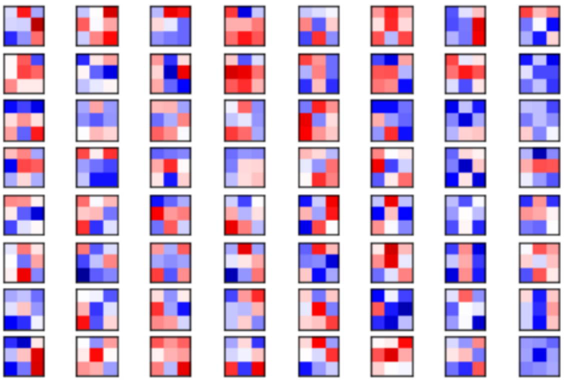

Figure 9. Filters of the third convolutional layer

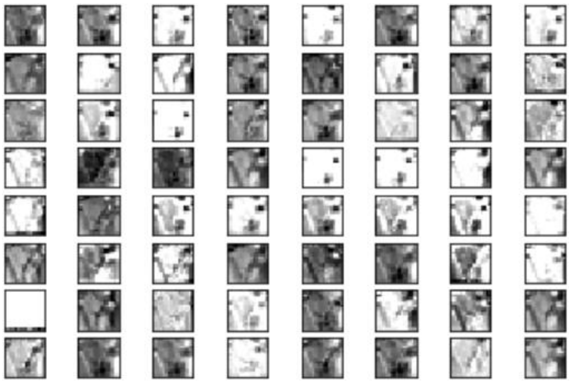

Figure 10. Feature maps generated by the third convolutional layer

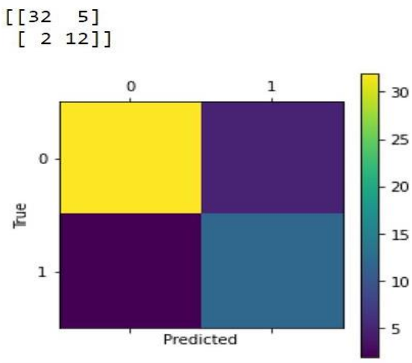

Figure 11. Confusion matrix of the best performance obtained

5. CONCLUSIONS

Multiple convolutional layers are recommended when it comes to CNNs for image classification. In this paper, the performance of the neural network decreased or did not show significant gains when more than 3 layers were used. The combination of a relatively low learning rate with the max pooling technique showed significant improvements, mainly because the network analyzed 128×128 px images. Even with the database being lacking in more images that faithfully represents slope structures and possible instabilities in them, the developed CNN obtained very satisfactory results, showing itself to be promising for the analysis of different types of structures and needs of the structure health monitoring (SHM) area. Among the main difficulties that were faced during the development of this paper, the following stand out: lack of a greater variety of images and achieving better results given the hardware limitations.

6. FUTURE WORKS

As future works, the main goal is to train this exact CNN architecture with a more robust dataset, that being with more than 2,000 slope and erosion images, to verify if its performance remains satisfactory or changes would be needed so that the network could become more accurate. Even though it is possible to develop a CNN application to analyze the progressive failure process of slopes, it would require an extensive dataset with several images of the same slope at different angles and timestamps, so that the factor of safety calculation could be accurate. Another good suggestion is to merge this CNN with an Autoencoder, so that the software would be able to predict and simulate how each kind of instability affects a slope structure over time, and not just detect the already existing ones.

ACKNOWLEDGEMENT

This work was supported by the Coordination for the Improvement of Higher Education Personnel—CAPES, the National Council for Scientific and Technological Development—CNPq, the Norte Energia S.A, and the support program for Qualified Production—PROPESP/UFPA (PAPQ) (notice 08/2022). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

REFERENCES

[1] H. Moayedi, A. Osouli, H. Nguyen, and A. S. A. Rashid, "A novel Harris hawks' optimization and k-fold cross-validation predicting slope stability," *Engineering with Computers*, vol. 37, no. 1, pp. 369–379, 2021, doi: 10.1007/s00366-019-00828-8.

[2] J. Poesen, "Soil erosion in the Anthropocene: research needs," *Earth Surface Processes and Landforms*, vol. 43, no. 1, pp. 64–84, 2018, doi: 10.1002/esp.4250.

[3] P. Borrelli *et al.*, "Soil erosion modelling: a global review and statistical analysis," *Science of The Total Environment*, vol. 780, p. 146494, Aug. 2021, doi: 10.1016/j.scitotenv.2021.146494.

[4] J. A. C. Calle, "Analysis of slope failure in unsaturated soil (in portugues : análise de ruptura de talude em solo não saturado)," University of São Paulo, 2000.

[5] Y. Wang *et al.*, "A CNN-based adaptive surface monitoring system for fused deposition modeling," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2287–2296, Oct. 2020, doi: 10.1109/TMECH.2020.2996223.

[6] X. Yang, Y. Gao, C. Fang, Y. Zheng, and W. Wang, "Deep learning-based bolt loosening detection for wind turbine towers," *Structural Control and Health Monitoring*, vol. 29, no. 6, Jun. 2022, doi: 10.1002/stc.2943.

[7] S. Liu, Z. Su, M. Li, and L. Shao, "Slope stability analysis using elastic finite element stress fields," *Engineering Geology*, vol. 273, p. 105673, Aug. 2020, doi: 10.1016/j.enggeo.2020.105673.

[8] C. Qi and X. Tang, "Slope stability prediction using integrated metaheuristic and machine learning approaches: a comparative study," *Computers and Industrial Engineering*, vol. 118, pp. 112–122, 2018, doi: 10.1016/j.cie.2018.02.028.

[9] J. Lu, L. Tan, and H. Jiang, "Review on convolutional neural network (CNN) applied to plant leaf disease classification," *Agriculture*, vol. 11, no. 8, p. 707, Jul. 2021, doi: 10.3390/agriculture11080707.

[10] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on convolutional neural networks (CNN) in vegetation remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 24–49, Mar. 2021, doi: 10.1016/j.isprsjprs.2020.12.010.

[11] S. S. Yadav and S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," *Journal of Big Data*, vol. 6, no. 1, p. 113, Dec. 2019, doi: 10.1186/s40537-019-0276-2.

[12] J. Chelliah, M. Alagarsamy, K. Anbalagan, D. Thangaraju, E. S. Wesley, and K. Suriyan, "Automatic wireless health instructor for schools and colleges," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 1, pp. 278–285, 2022, doi: 10.11591/eei.v11i1.3330.

[13] Liliana, J. H. Chae, J. J. Lee, and B. G. Lee, "A robust method for VR-based hand gesture recognition using density-based CNN," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 2, pp. 761–769, 2020, doi: 10.12928/TELKOMNIKA.v18i2.14747.

[14] Y. F. Tan, X. Guo, and S. C. Poh, "Time series activity classification using gated recurrent units," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 4, pp. 3551–3558, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3551-3558.

[15] A. E. Minarno, W. A. Kusuma, and Y. A. Kurniawan, "Human activity recognition for static and dynamic activity using convolutional neural network," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 6, p. 1857, Dec. 2021, doi: 10.12928/telkomnika.v19i6.20994.

[16] O. Dahmane, M. Khelifi, M. Beladgham, and I. Kadri, "Pneumonia detection based on transfer learning and a combination of VGG19 and a CNN built from scratch," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 3, pp. 1469–1480, Dec. 2021, doi: 10.11591/ijeecs.v24.i3.pp1469-1480.

[17] X.-K. Dang, L. A.-H. Ho, X.-P. Nguyen, and B.-L. Mai, "Applying artificial intelligence for the application of bridges deterioration detection system," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 20, no. 1, p. 149, Feb. 2022, doi: 10.12928/telkomnika.v20i1.20783.

[18] H. A. Saeed, S. Hamad, and A. T. Hussain, "Analysis the digital images by using morphology operators," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 3, pp. 1654–1662, 2021, doi: 10.11591/ijeecs.v24.i3.pp1654-1662.

[19] M. Mishra, V. R. Gunturi, and T. F. D. S. Miranda, "Slope stability analysis using recent metaheuristic techniques: a comprehensive survey," *SN Applied Sciences*, vol. 1, no. 12, p. 1674, Dec. 2019, doi: 10.1007/s42452-019-1707-6.

[20] E. Sharifi Teshnizi, M. Golian, S. Sadeghi, and A. Rastegarnia, "Application of analytical hierarchy process (AHP) in landslide susceptibility mapping for Qazvin province, N Iran," in *Computers in Earth and Environmental Sciences*, Elsevier, 2022, pp. 55–95.

[21] M. Ließ, B. Glaser, and B. Huwe, "Functional soil-landscape modelling to estimate slope stability in a steep Andean mountain forest region," *Geomorphology*, vol. 132, no. 3–4, pp. 287–299, 2011, doi: 10.1016/j.geomorph.2011.05.015.

[22] M. T. Löbmann, C. Geitner, C. Wellstein, and S. Zerbe, "The influence of herbaceous vegetation on slope stability – a review," *Earth-Science Reviews*, vol. 209, p. 103328, Oct. 2020, doi: 10.1016/j.earscirev.2020.103328.

[23] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, Dec. 2021, doi: 10.1186/s40537-021-00444-8.

[24] Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, "Deep learning for remote sensing image classification: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 6, p. e1264, Nov. 2018, doi: 10.1002/widm.1264.

[25] G. W. Lindsay, "Convolutional neural networks as a model of the visual system: past, present, and future," *Journal of Cognitive Neuroscience*, vol. 33, no. 10, pp. 2017–2031, Sep. 2021, doi: 10.1162/jocn_a_01544.

[26] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: an overview," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6690–6709, 2019, doi: 10.1109/TGRS.2019.2907932.

[27] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, Aug. 2017, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186.

[28] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, vol. 49, no. 6, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[29] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, vol. 40, no. 1, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[30] N. Akhtar and U. Ragavendran, "Interpretation of intelligence in CNN-pooling processes: a methodological survey," *Neural Computing and Applications*, vol. 32, no. 3, pp. 879–898, Feb. 2020, doi: 10.1007/s00521-019-04296-5.

[31] N. Murray and F. Perronnin, "Generalized max pooling," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 2473–2480, doi: 10.1109/CVPR.2014.317.

[32] B. Graham, "Fractional max-pooling," *arXiv preprint arXiv:1412.6071*, pp. 1–10, 2014, doi: 10.48550/arXiv.1412.6071.

[33] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, vol. 2019-June, pp. 558–567, doi: 10.1109/CVPR.2019.00065.

[34] J. Nagi *et al.*, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, Nov. 2011, pp. 342–347, doi: 10.1109/ICSIPA.2011.6144164.

[35] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," *arXiv preprint arXiv:1511.05879*, pp. 1–12, Nov. 2015, doi: 10.48550/ARXIV.1511.05879.

[36] S. H. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," *Neurocomputing*, vol. 378, pp. 112–119, Feb. 2020, doi: 10.1016/j.neucom.2019.10.008.

[37] G. B. B. Bademian, "Classification of traffic signs with neural networks (in portuguese: classificação de placas de trânsito com redes neurais)," Federal University of São Carlos, 2022.

[38] I. V. Rizzo and R. L. C. Canato, "Artificial intelligence: activation functions (in portuguese: inteligência artificial: funções de ativação," *Prospectus (ISSN: 2674-8576)*, vol. 2, no. 2, pp. 51–65, 2020.

[39] T. S. Verne, F. A. da Silva, L. L. de Almeida, D. R. Pereira, and A. O. Artero, "Detection and recognition of small plants using machine learning (in portuguese: Detecção e reconhecimento de plantas de pequeno porte utilizando aprendizagem de máquina)," *Colloquium Exactarum*, vol. 14, no. 1, pp. 36–45, 2022, doi: 10.5747/ce.2022.v14.n1.e383.

[40] Z. Zhang, "Improved Adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, Jun. 2018, pp. 1–2, doi: 10.1109/IWQoS.2018.8624183.

[41] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR*, Dec. 2014, pp. 1–15, doi: 10.48550/arXiv.1412.6980.

[42] S. Bock, J. Goppold, and M. Weiß, "An improvement of the convergence proof of the ADAM-optimizer," *arXiv preprint arXiv:1804.10587*, pp. 1–5, 2018, doi: 10.48550/ARXIV.1804.10587.

[43] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: an empirical study of their impact to deep learning," *Multimedia Tools and Applications*, vol. 79, no. 19–20, pp. 12777–12815, 2020, doi: 10.1007/s11042-019-08453-9.

[44] W. G. Chang, T. You, S. Seo, S. Kwak, and B. Han, "Domain-specific batch normalization for unsupervised domain adaptation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, vol. 2019-June, pp. 7346–7354, doi: 10.1109/CVPR.2019.00753.

[45] T. Ullmann, C. Hennig, and A. Boulesteix, "Validation of cluster analysis results on validation data: a systematic framework," *WIREs Data Mining and Knowledge Discovery*, vol. 12, no. 3, May 2022, doi: 10.1002/widm.1444.

[46] R. M. Watanabe, "Summer rains, stay tuned (in portuguese: Chuvas de Verão, fique ligado)," *Civil Defense of Duque de Caxias*, 2017. http://defesacaxias.blogspot.com/2011/12/chuvas-de-verao-fique-ligado.html (accessed Jun. 02, 2022).

[47] "Kaki seedlings-floriculture (in portuguese: Kaki mudas-Floricultura)," *Kaki Mudas*, 2017. https://kaki-mudas.negocio.site/ (accessed May 11, 2022).

[48] C. Aroh, "Civil Defense, community struggle to rescue Anambra road from erosion," *The Whistler*, 2022. https://thewhistler.ng/civil-defence-community-struggle-to-rescue-anambra-road-from-erosion/ (accessed Jun. 02, 2022).

BIOGRAPHIES OF AUTHORS

Caio Henrique Esquina Limão Graduated from the University Center of the State of Pará (CESUPA) in 2019 in the Computer Engineering course. Master's student in Electrical Engineering at the Federal University of Pará (UFPA), in the area of applied computing. Has experience on the development of systems and softwares applying computational intelligence and integrated IoT solutions, as well as in web development. He can be contacted at email: caiolimao97@gmail.com or caio.limao@itec.ufpa.br.

Thabatta Moreira Alves de Araújo Graduated from the Pontifical Catholic University of Minas Gerais (2011) in the Control and Automation Engineering course. Master in Civil Engineering from the Federal University of Alagoas (UFAL), area of concentration in Structures and pursuing a PhD in Electrical Engineering at UFPA. Experience in micro automation projects, electronics and instrumentation, development of systems and software in computational intelligence. He worked as a professor at the University of Uberaba, in Parauapebas (PA) in electrical engineering and civil engineering courses. Worked as a professor at the Federal Institute of Science and Technology Education - IFPA, Parauapebas campus, where she also coordinated the higher course of Technology in Industrial Automation. Is currently a professor at the Department of Informatics, Management and Design at the Federal Center for Technological Education of Minas Gerais (CEFET-MG), Divinópolis campus. She can be contacted at email: thabatta@cefetmg.br.

Carlos Renato Lisboa Francês 1-D Research Productivity Scholarship. Completed a post-doctoral internship at the Institute of Systems and Computer Engineering, Technology and Science (INESC-TEC), Porto/Portugal, 2012. Holds a Bachelor's degree in Computer Science from the Federal University of Pará (1995), a Master's degree in Computer Science and Computing and Computational Mathematics from the University of São Paulo (1998) and a PhD in Computer Science and Computational Mathematics from the University of São Paulo (2001). He is a professor at the Faculty of Computer Engineering and Telecommunications, at the Federal University of Pará. He was president of the Data Processing Company of the State of Pará (from 01/2007 to Dec/2010), being responsible for the implementation of the NavegaPará digital inclusion program. Until 2011, he was a member of the first Advisory Committee on Computing of the National Institute of Educational Studies and Research (INEP-MEC). He was the first Dean of Graduate Studies, Research and Technological Innovation at the Federal University of the South and Southeast of Pará (Unifesspa), from 2013 to 2016, was the Pro-Tempore Dean of the Federal University of the South and Southeast of Pará (Unifesspa), from June to October 2016. He is currently Director of the Technological Innovation Agency at UFPA. He advises the Graduate Programs in Electrical Engineering (PPGEE) and Science. of Computing (PPGCC), both from UFPA, having graduated 34. He can be contacted at email: rfrances@ufpa.br.