Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Enhancing Agricultural Efficiency: Deep Learning-Based Soil Crack Detection for Water Irrigation

Khin Moe Myint[1], Maung Aye[2] , Tin Tin Hla[3]
[1]khinmoemyinttoe@gmail.com , [2]kokoaye528@gmail.com, [3]tintinhla99@gmail.com
[1,2,3] Mandalay Technological University, Myanmar,Department of Electronic Engineering

Article Information

Submitted : 7 May 2024
Reviewed : 10 May 2024
Accepted : 15 Jun 2024

Keywords

Dataset, Inception V3,Raspberry Pi, Soil Crack Detection

Abstract

The escalating demand for agricultural precision and environmental monitoring underscores the necessity for effective soil crack detection methods. This study explores the feasibility of employing a Raspberry Pi-powered camera system and deep learning image recognition to detect soil cracks and control agricultural irrigation. The purpose is to develop a soil crack detection system using deep learning techniques, sustain plant growth process, increase productivity, and optimize water irrigation practice. Our approach leverages TensorFlow to craft a convolutional neural network tailored specifically for execution on a Raspberry Pi 3B+. A dataset comprises manually captured images and is trained with the InceptionV3 model categorized into crack or nocrack classes. The accuracy is achieved ranging from 97% to 99%. These results underscore deep learning image recognition models on Raspberry Pi for cost-effective soil crack monitoring and controlling the plants watering system.

A. Introduction

The soil crack formation are multifaceted and can be attributed to various natural processes and human activies factors. Natural factors like biotic processes, hypergenesis and geological phenomena are significant contributors. Hypergenesis exacerbating soil cracking especially in arid regions with high salt levels. Biotic activities, such as plant root growth and burrowing organisms, can disrupt soil integrity. Geological processes, shape the Earth's surface and influence soil stability. Hydrogeological processes, driven by underground water movement, contribute to soil erosion and deposition. Both natural and man-made earthquakes impact soil stability and surface features. Additionally, human activities like intensive agriculture, deforestation, and urbanization worsen soil degradation, increasing susceptibility to cracking[1]. Comprehending the various factors contributing to soil cracking is crucial for successful soil management and conservation efforts.[2] Cracks naturally due to moisture evaporation and reduced water content in soil, while swelling occurs in expansive soils with higher water content. Exposure to atmospheric conditions alters the behavior of expansive soils like clays or mad, influenced by local weather fluctuations. Cracks can occur due to the thermo-hydromechanical (THM) problem, which involves the water losing through evaporation and the problem of fracture mechanics (FM) [3].

. Researchers are exploring its potential for image recognition tasks, promising cost-effective solutions for computer vision applications. Introduced in 2012, the Raspberry Pi has become a prominent single-board computer known for its versatility and affordability. Widely utilized in computer science education, it offers practical hands-on learning experiences. [7-11] Moreover, it plays an essential role in the IoT as a gateway device, connecting sensors, actuators, and other IoT devices to the internet, facilitating seamless communication within IoT ecosystems. In [12] Raspberry Pi, captured images are sent to the cloud for image recognition. Unlike traditional methods, we employ a convolutional neural network (CNN) for our model. Although CNNs have gained immense popularity in image classification [13], there is a notable scarcity of research on implementing CNNs on Raspberry Pi, predominantly owing to the restricted processing power of older Raspberry Pi iterations. The analysis revealed that employing a CNN framework that integrated TensorFlow [14], [15] resulted in the optimal performance on the Raspberry Pi. Among the various CNN models, [5] Inception V3 has garnered significant attention for its superior performance in image classification tasks. Its complex architecture enables comprehensive feature extraction, making it well-suited for detecting subtle patterns in soil cracks. It can use in soil crack detection holds promise for understanding soil degradation processes and implementing timely intervention measures. The importance of efficient resource management in agricultural production, historically reliant on natural resources, is paramount. The analysis of agricultural water distribution systems entails two fundamental rules, where the distribution methods encompass both manual (hand watering) and automated (automatic watering) systems. The automatic watering systems include tube watering, capillary mat watering, overhead sprinklers, and drip irrigation mechanisms.

In this paper, we utilize freely available components to emulate conservationists' practices, such as the Raspberry Pi, reducing costs for deploying camera sensor systems. We select the Raspberry Pi Model 3B+ for its features and compatibility with TensorFlow, chosen for their user-friendly interfaces and machine learning capabilities. The contributions of this research paper are outlined as follows:

- A thorough investigation of prominent literature on soil crack detection utilizing machine learning and deep learning methodologies is undertaken.
- The introduction of a soil crack detection problem is proposed, employing a pre-trained Inception V3 Convolutional Neural Network (CNN) model for classifying various soil crack detections.
- A comprehensive soil dataset is meticulously compiled and organized.
- The proposed model is trained with a CPU to achieve optimal accuracy and minimal validation loss, significantly enhancing its suitability for integration into Raspberry Pi applications, thereby strengthening their capabilities in precise data processing and decision-making tasks.
- The required libraries and software frameworks are installed, utilizing TensorFlow backend for model development and integration with the Raspberry Pi.
- Finally, an automatic watering system is implemented and driving the pump motor control system based on the detected crack condition.

The remaining sections of this paper are in section 2 elucidates the pertinent knowledge and outlines the proposed architecture for crack detection using Deep learning model, while section 3 outlines the dataset collection preprocessing and methodology utilized. Section 4 conducts experimental validation, showcasing outcomes from the proposed model alongside analogous comparative models and a detailed discussion is presented on the obtained results. Finally, Section 5 provides a summary of the main points and concludes of this study.

B. Related Works

Researchers have primarily concentrated on studying soil crack development and morphology, along with quantitative measurements to comprehend crack patterns [16]. [17] have also employed diverse methods to analyze, predict and quantify crack formation in expansive soils. Despite limitations in traditional crack detection methods, newer methods utilizing modern technologies have spurred development. Within Digital Image Processing, Machine Learning and Deep Learning algorithms are prevalent methods for crack detection and recognition. Machine learning-based methods, such as support vector machine (SVM) [18] and extreme learning machine (ELM) [19], are utilized to classify cracks in concrete from input images, enabling the handling of large datasets. Additionally, Andrushia et al. [20] explored crack detection techniques for concrete structures exposed to fire, analyzing images in the transformed domain to identify and pinpoint crack locations.In various machine learning approaches, a crucial initial step involves extracting features to understand input images. Feature reduction techniques are then applied to optimize feature selection, often employing Principle Component Analysis (PCA) for informative feature sets. These selected features are fed into classifiers, such as K Nearest Neighbor (KNN), Artificial Neural Networks (ANN), Fuzzy Logic with Support Vector Machine (SVM), and Genetic Algorithms with

SVM, to accurately detect cracks in input images [21]. Traditionally ,pattern recognition involves pre-processing, feature extraction, segmentation, classification, and recognition, each stage offering opportunities for algorithmic intervention. However, feature extraction presents a challenge as it depends on user-provided high-level features, possibly compromising detection accuracy.This challenge is addressed by employing deep learning techniques capable of automatic feature extraction and handling large input volumes.

Deep learning, a rapidly advancing artificial intelligence tool, has revolutionized pattern recognition, showcasing superior performance compared to traditional machine learning methods [22]. These methods leverage convolutional neural networks (CNNs) to extract intricate spatial and textural features from soil images, enabling accurate crack identification and quantification. Several studies have explored the efficiency of deep learning techniques, including CNN architectures, for soil crack detection, demonstrating promising results. Researchers have effectively employed deep convolutional neural networks (CNNs) for concrete crack detection [23], utilizing architectures like U-net with residual connections. For pavement crack detection, ensemble networks and encoder-decoder architectures have achieved 92.1% accuracy [24]. Deep learning models have also been utilized for concrete structure crack detection, comparing multiple pretrained models [25]. Additionally, four deep neural models, including customized CNN models, VGG-19, and ResNet-50, are used for soil shrinkage crack detection, showing improved accuracy over pretrained models. Determining foundation types relies on soil type accuracy, driving the development of deep learning-based approaches. In agriculture, deep learning methods such as U-net and FCN-8s are investigated for crop/weed discrimination, with FCN-8s achieving 75.1% accuracy in detecting weeds and U-net performing better in detecting crops with 60.48% accuracy.The adaptability of Raspberry Pi system has also made it integral to robotics projects, serving as the computational backbone for controlling and coordinating robotic systems [26]. Moreover, in sensor networks, the Raspberry Pi offers a convenient platform for collecting, processing, and transmitting data from distributed sensor nodes, facilitating diverse sensing applications [27][28].

Our study investigates the Raspberry Pi's effectiveness in detecting soil cracks using a pre-trained Inception V3 model for automatic tube water irrigation systems. Leveraging TensorFlow and transfer learning is trained on images from the Pi camera module, aiming to detect soil crack images efficiently.

C. Research Method

Dataset Creating

To detect the soil crack for agricultural water distribution approach,we employed transfer learning to develop a CNN of deep learning algorithm. The transfer learning model adeptly utilizes images from our proprietary dataset while also capitalizing on established datasets such as ImageNet [29] and VIPeR [30]. The ImageNet dataset, curated by Stanford University, categorizes images into classes. The resource of ImageNet dataset simplifies the process of accessing, downloading, and utilizing a substantial pool of images for supervised learning tasks. In contrast, our custom dataset consists of soil images captured in various agricultural regions of Myanmar, encompassing diverse perspectives and lighting conditions. In this system, a dataset comprising 500 images is established in each categories, with crack images labeled as 'crack' and uncrack images as 'no crack'. All images are captured using a Pi camera with a specific resolution. The dataset, comprising 1000 images, is subsequently utilized for training, testing, and validation purposes. Preprocessing is essential to enhance image analysis accuracy, achieved through Python OpenCV to resize images to 299x299x3, facilitating better feature detection for tracks, cracks, and scratches. This dataset is provided to assess model performance in contemporary surveillance systems.

Inception v3

Inception v3 is utilized for classification and recognition tasks in this paper. Deep learning frameworks, categorized into non-symbolic (e.g., Torch and Caffe) and symbolic (e.g., TensorFlow, Theano), offer distinct advantages. While non-symbolic frameworks require manual optimization and are less easily modified, symbolic frameworks automatically optimize and exploit memory reuse more effectively. The versatility of Inception V3 is evident in its capacity to adapt vast datasets and accommodate images of different sizes and resolutions. This flexibility is particularly significant within the medical imaging field, where images can vastly differ in their dimensions, resolutions, and overall quality. Typically, the Inception, as detailed by Ting et al. in 2017, incorporates a combination of one maximum pooling layer and three convolution layers of varying sizes. This module serves to aggregate channels from the preceding layer, followed by a non-linear fusion operation. In this specific model, the inclusion of the Inception module ensures that over fitting is mitigated while simultaneously enhancing the network's expressive capability and adaptability to diverse scales. In the Inception-v3 procedure, a transfer learning approach is employed. It, presented in Equation 1, is a pre-trained deep convolutional neural network consisting of 48 layers. It excels in the task of learning and identifying intricate patterns and features within crack images. Its primary capability lies in its adeptness at discerning complex patterns and features present in crack images.

$$\mathrm{A_x} = \begin{pmatrix} A_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,n} \end{pmatrix} * \begin{pmatrix} B_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ B_{m,1} & \cdots & B_{m,n} \end{pmatrix} \tag{1}$$
$$= \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} A_{(M-1),(N-1)} B_{(i+1),(j+1)}$$

In order to prevent over fitting, the output layer is flattened and reduced its dimensions into one. Subsequently, a sigmoid layer was incorporated for classification, along with a fully connected layer containing 1,024 hidden units. We employed a Relu activation function, denoted as Equation 2, and applied dropout with a rate of 0.4 to further enhance performance.

$$f(x) = \max(0, x) \tag{2}$$

For initializing the weights of the classification layers, we followed the algorithm outlined in (Beauxis-Aussalet and Hardman, 2014), presented in

Equation 3. This strategy effectively enabled us to utilize Inception V3 for our specific objectives.

$$Wk \sim U(-\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}}) \quad (3)$$

The parameters of the preceding layer are retained while the last layer is removed. The soil dataset is then introduced to retrain the new last layer, adjusting the number of output nodes to two different soil crack categories. By employing the backpropagation algorithm, the last layer of the model can be trained effectively, and the weight parameters are adjusted by calculating the error between the output of the SoftMax layer and the label vector corresponding to the given sample category using cross-entropy of the cost function [31] .

System Design

The main steps included in the methodology of this system encompass dataset creation, preprocessing, and training, testing, and validation. Dataset creation is fundamental, necessitating the generation of a comprehensive dataset for accurate outcomes. Figure 1 shows the block diagram of overall system.

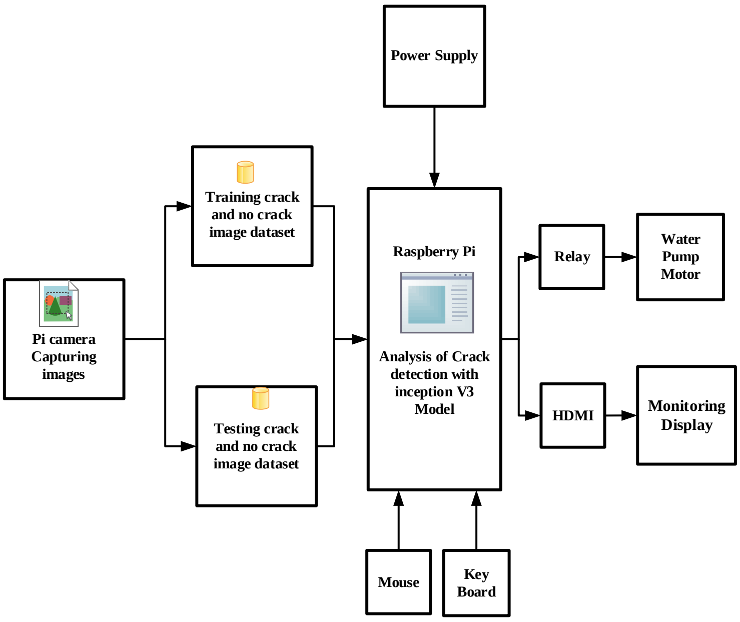


Figure 1. Block Diagram of Overall System Design

Once the necessary components, including the Raspberry Pi 3 B+, SMP 0023 Pi camera module, HDMI cable,7inches display monitor, external keyboard, mouse, and power supply, are selected. They are interconnected with the Raspberry Pi. It undergoes installation of the operating system and configuration of modules to facilitate the operation of the image classification system. This next setup, the pre-trained and tested program code from the computer. If the simulation performance on CPU obtained higher accuracy, it would be deployed onto the Raspberry Pi. Finally, training, testing, and validation are conducted utilizing TensorFlow learn package in Python and the Inception v3 model, ensuring robust performance in crack detection. Additionally, the relay and water pump are integrated with the Raspberry Pi to establish the water distribution system. As the image classification system achieves satisfactory accuracy results, the relay is triggered, initiating the continuous operation of the water pump for efficient water distribution. Figure 2 shows the flowchart of this system.

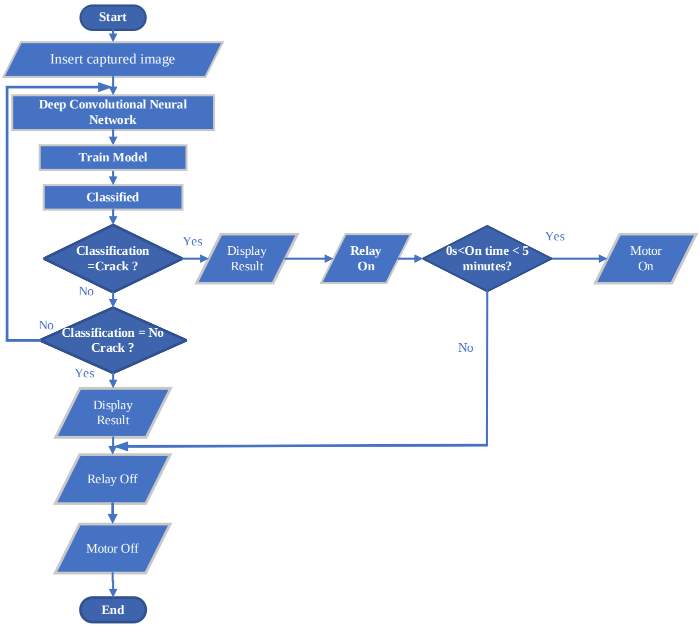


Figure 2. System FlowChart

Plants Watering System

After soil crack has been detected, the automatic watering system will initiate its operation. In the event that the soil is identified as cracked, the water pump will activate and run for approximately 5 minutes, ensuring sufficient hydration to mitigate soil degradation. Conversely, if the soil is recognized as intact with no cracks, the water pump will remain off, conserving resources and minimizing unnecessary irrigation. This automated process optimizes water usage by delivering targeted hydration precisely where it is needed, thereby promoting efficient and sustainable plant growth practices.

D. Result and Discussion

This research is divided into two sections, aiming an optimal model for accurately identifying and classifying different soil types based on crack and continuously followed by an exploration of its real-world applications. There are two sections as software and hardware implementation on CPU and Raspberry Pi Computer.

Experimental Setup in CPU

Data are collected as capturing crack and no crack images using a Pi camera, resulting in a dataset comprising 600 crack images and 600 no crack images organized into subfolders for training purposes. Subsequently, a testing dataset is curated with 60 crack images and 60 no crack images, similarly organized into subfolders for evaluation. The dataset needs manual screening, and each soil crack target must be labeled. To meet the experiment's criteria, the soil image dataset was divided into training, testing, and validation sets at an 8:1:1 ratio for model testing. The experimental setup on CPU is listed in Table 1. Table 1 highlights important parameter settings used in training. It primarily employs the default stochastic gradient descent (SGD) method for optimization. However, it also mentions the option to use the adaptive moment estimation (Adam) algorithm depending on the training conditions.

Table 1. Experimental Setup

No	Enviromental Lab	Detail
1	Operating System	Window 10
2	Programming language	Python 3.7.5
3	Framework of Deep Learning	TensorFlow

Table 2. Experimental Setup

No	Training Parameters	Detail
1	Image size	299x299x3
2	epcho	2000
3	Batch size	16
4	Learning rate	0.01

The software implementation utilizes a pre-trained deep convolutional network to facilitate crack detection, employing a systematic approach to dataset construction and testing. The simulation results entail training the dataset with the Inception V3 model within the TensorFlow framework, augmented by visualizations of the training model graph through Tensor Board. Figure 3 illustrates the visual representation of the soil crack or no soil crack classification model graph on Tensor Board, providing insights into the network's architecture.

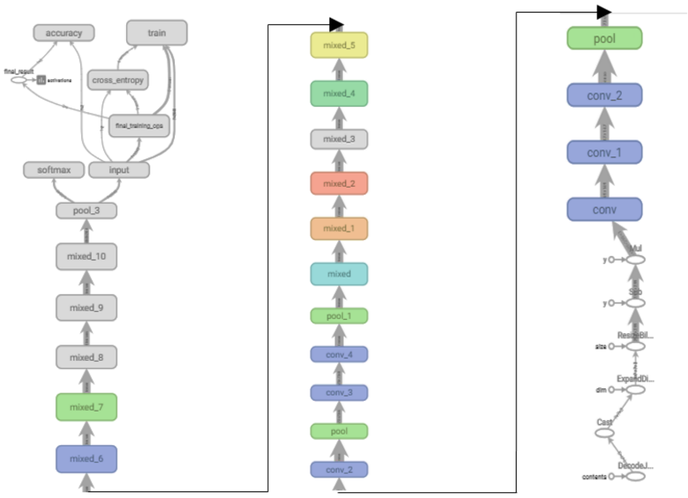
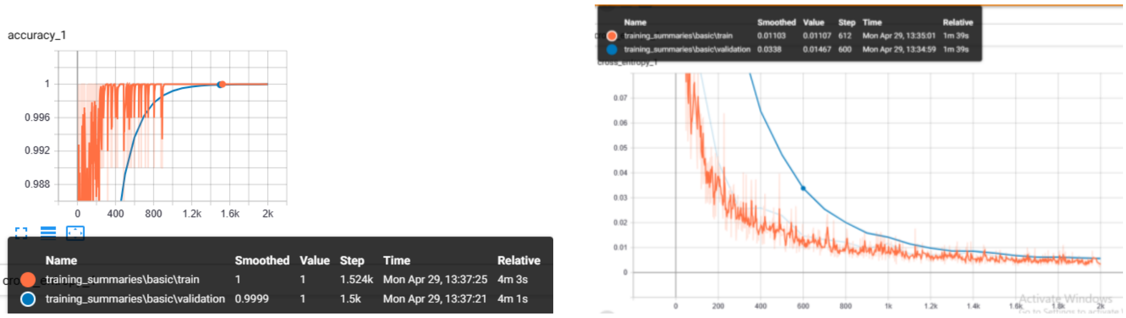


Figure 3. Visualizing of Soil Crack Classification Model Graph on Tensor Board

Performance Analysis in Training of Image Classification

Performance analysis is conducted by assessing image classification through metrics such as training accuracy and cross-entropy, as depicted in Figure 4 (a) and (b) showcasing the cross-entropy graph of the trained model. The accuracy reach over 0.98 at the first 10 epcho and then fewer flucution between 0.995 and 0.997 accuracy from epcho 20 to 70. Until epcho 100 suddently drop to 0.988 accuracy. The accuarcy of graph fall and rise bounce nearly 1 from the next epcho to epcho 900. After the epcho 900, the training model of accuracy stable to 1. The cross-entropy of training model get 0.08 at frst 50 epcho but the valdation graph obtain the same value at epcho 370. The more increase the next epcho step by step, the more decrease the loss function of training and valadition of model training.

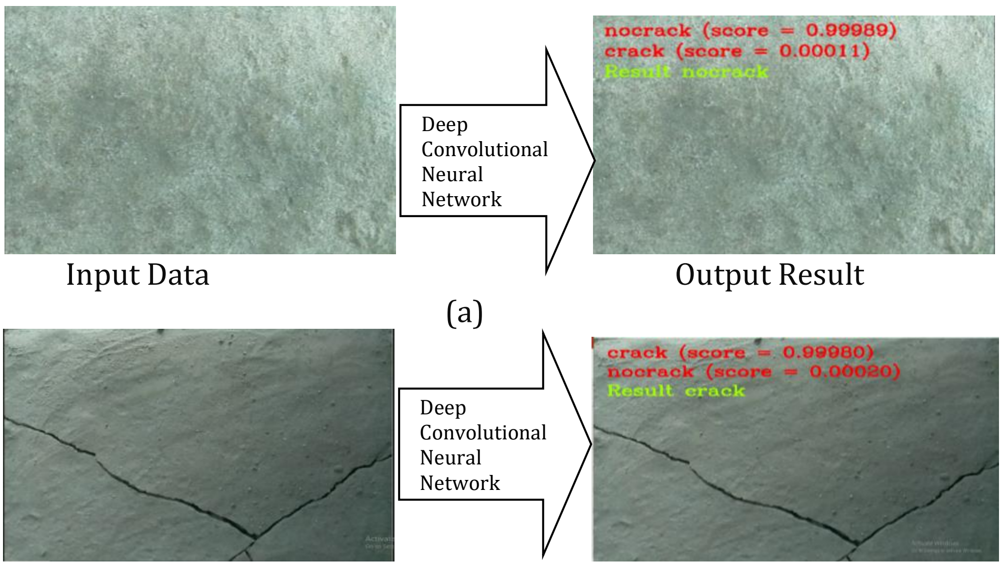


(a) (b)

Figure 4. (a) Accuracy Graph, (b) Cross Entropy Graph of our training dataset with Inception V3 Model

Testing

Finally, the testing dataset is subjected to simulation using the Inception V3 model in TensorFlow, enabling comprehensive evaluation of the software's performance on a computer-based platform. As the test image data is fed into the deep learning model, the resulting output will be classified as either containing a crack or not as shown in Figure 5(a) and (b).



Input Data Output Result

(b)

Figure 5. Simulation result of test dataset for (a) no crack (b) Crack

Experimental Setup on Raspberry Pi Computer

Testing on Raspberry Pi Computer

Dataset images are meticulously collected using the Pi camera, capturing instances of both crack and no crack scenarios. Following the implementation of our pre-trained model within the Raspberry Pi environment, comprehensive training, testing and analysis are conducted to evaluate its efficacy.

In Figure 6, the output results exhibit accurate classification as they pass through our system, demonstrating the model's capability to correctly identify crack images. Similarly, Figure 7showcases the successful classification of input no crack test images captured by the Pi camera, reaffirming the robustness and reliability of our system in distinguishing between crack and no crack instances with their respective precision percentage. These results underscore the effectiveness of our pre-trained model in real-world applications, offering promising insights into its potential for practical deployment in crack detection systems.

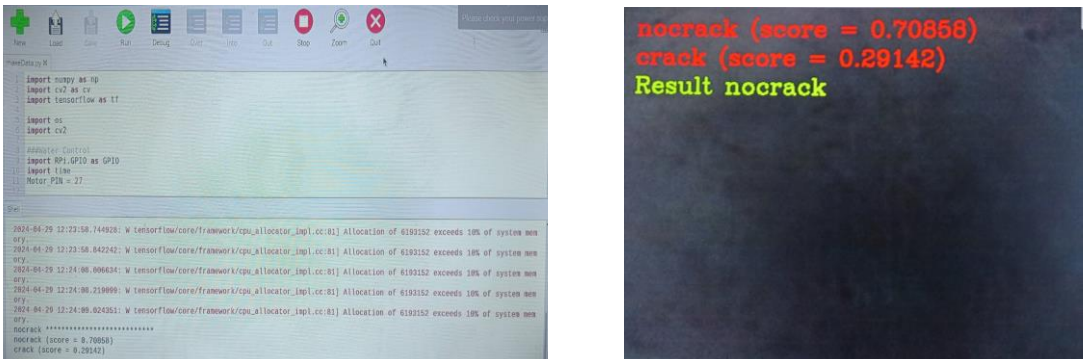


Figure 6. Experimental Result of No Crack Image Classification

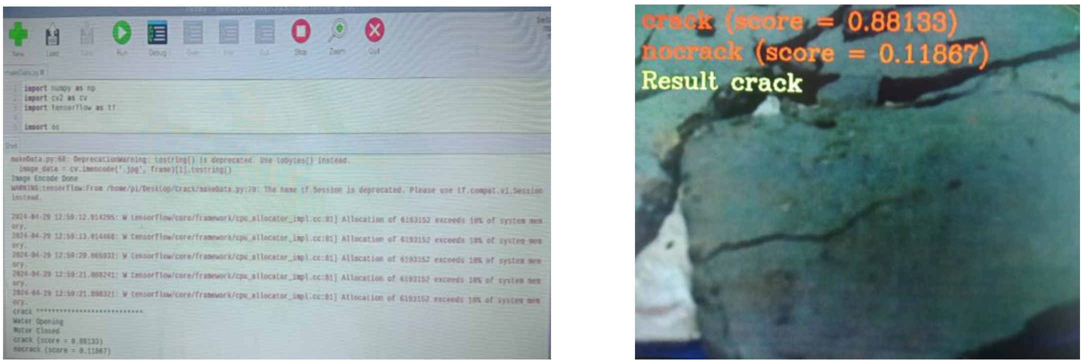


Figure 7. Experimental Result of Crack Image Classification

Performance Evaluation

The performance evaluation of soil crack classification with our pre-trained deep learning model is assessed using the accuracy, calculated as the ratio of correctly classified images to the total number of images. Specifically, accuracy is computed using the formula:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN),$$

where ,
TP= the number of soil crack images correctly identified as soil crack,
TN=the number of no soil crack images correctly identified as no soil crack,
FP=the number of soil crack images incorrectly classified as no soil crack,
FN=the number of no soil crack images incorrectly classified as soil crack

In our evaluation, the accuracy of our soil crack classification model is determined to be 99.7%, reflecting the model's ability to effectively distinguish between soil crack and no soil crack instances with impeccable precision.

Discussion

Inception V3 is employed as a classifier, we conducted our experiments and implemented the model with cutting-edge alternatives, evaluating crack detection results both on CPU and Raspberry Pi Computer. Deep learning methods, while demanding longer training times due to their reliance on extensive operations and large datasets, offer superior accuracy compared to traditional machine learning approaches. Leveraging the Python programming language, the TensorFlow framework seamlessly supports the implementation of deep learning algorithms, providing a high-level interface that simplifies the image classification or recognition process. Importantly, the incorporation of a deep learning model does not escalate the overall cost of the system since our image recognition technology relies on freely available software and tools. Furthermore, the cost-effectiveness and efficiency of this water distribution system outperform those of pricier alternatives, rendering it ideal for agricultural applications.

E. Conclusion and Future Work

In this system, a Deep Convolutional Neural Network (CNN) serves as the cornerstone for extracting crucial features from images within the training dataset. The model of Inception v3 is employed to comprehend the underlying patterns within the training data and subsequently classify the test data accurately. The training of the model on Raspberry pi spanned 90 minutes across 12 epochs, averaging 18 minutes per epoch. Our experiments reveal a soil crack detection accuracy of at least 99 percent, with image classification occurring at acceptable speeds conducive to real-time processing. These outcomes suggest that a Raspberry Pi-based camera system, integrated with a deep learning model, can significantly aid conservation efforts. This system benefits conservationists exploring Raspberry Pi for monitoring soil cracks and water supply in agriculture.

In the future, we will explore additional optimization techniques, such as model quantization, integrating with IoT and cloud systems, and incorporating remote sensing control. Although these optimizations might result in a slight decrease in accuracy, they have the potential to significantly lower power consumption and memory requirements, ultimately improving the efficiency and scalability of our models.

F. Acknowledgment

This research initiative was undertaken within the esteemed laboratory facilities of the Department of Electronic Engineering at Mandalay Technological

University, Myanmar. I wish to convey my profound appreciation to my supervisor, U Maung Aye, for their invaluable guidance, and my head department, Dr Tin Tin Hla, unwavering support, and profound insights throughout this endeavor. Furthermore, I would like to extend my gratitude to the Department of Electronic Engineering at MTU for providing crucial resources, cutting-edge equipment, and comprehensive laboratory infrastructure, all of which played an integral role in the successful culmination of this study focused on soil crack classification to those who have provided support for the research.

G. References

[1] Xu, J.-J., Zhang, H., Tang, C.-S., Cheng, Q., Liu, B., & Shi, B. (2022). Automatic soil desiccation crack recognition using deep learning. *Géotechnique*, *72*(4), 337–349.*https://doi.org/10.1680/jgeot.20.P.091*.

[2] Andrushia, A. D., Neebha, T. M., Umadevi, S., Anand, N., & Cashell, K. A. (n.d.). Shrinkage Crack Detection in Expansive Soil using Deep Convolution Neural Network and Transfer Learning. In *KSCE Journal of Civil Engineering*.

[3] Yan, J., Downey, A., Cancelli, A., Laflamme, S., Chen, A., Li, J., & Ubertini, F. (2019). Concrete Crack Detection and Monitoring Using a Capacitive Dense Sensor Array. *Sensors*, *19*(8), 1843. *https://doi.org/10.3390/s19081843*

[4] Zhao, H., Qin, G., & Wang, X. (2010). Improvement of canny algorithm based on pavement edge detection. *2010 3rd International Congress on Image and Signal Processing*, 964–967. *https://doi.org/10.1109/CISP.2010.5646923*

[5] Andrushia, A. D., Neebha, T. M., Umadevi, S., Anand, N., & Cashell, K. A. (n.d.). Shrinkage Crack Detection in Expansive Soil using Deep Convolution Neural Network and Transfer Learning. In *KSCE Journal of Civil Engineering*.

[6] Ellison Mathe, S., Bandaru, M., Kishan Kondaveeti, H., Vappangi, S., & Sanjiv Rao, G. (2022). A Survey of Agriculture Applications Utilizing Raspberry Pi. *2022 International Conference on Innovative Trends in Information Technology (ICITIIT)*, 1–7. *https://doi.org/10.1109/ICITIIT54346.2022.9744152.*

[7] Johnston, S., & Cox, S. (2017). The Raspberry Pi: A Technology Disrupter, and the Enabler of Dreams. *Electronics*, *6*(3), 51. *https://doi.org/10.3390/electronics6030051*

[8] Vappangi, S., Penjarla, N. K., Mathe, S. E., & Kondaveeti, H. K. (2022). Applications of Raspberry Pi in Bio-Technology: A Review. *2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP)*, 1–6. *https://doi.org/10.1109/AISP53593.2022.9760691*

[9] M. Hosny, K., Magdi, A., Salah, A., El-Komy, O., & Lashin, N. A. (2023). Internet of things applications using Raspberry-Pi: a survey. *International Journal of Electrical and Computer Engineering (IJECE)*, *13*(1), 902. *https://doi.org/10.11591/ijece.v13i1.pp902-910*

[10] Wilkinson, M., Bell, M. C., & Morison, J. I. L. (2021). A Raspberry Pi-based camera system and image processing procedure for low cost and long-term monitoring of forest canopy dynamics. *Methods in Ecology and Evolution*, *12*(7), 1316–1322. *https://doi.org/10.1111/2041-210X.13610*

[11] Mathe, S. E., Pamarthy, A. C., Kondaveeti, H. K., & Vappangi, S. (2022). A Review on Raspberry Pi and its Robotic Applications. *2022 2nd International*

Conference on Artificial Intelligence and Signal Processing (AISP), 1–6. *https://doi.org/10.1109/AISP53593.2022.9760590*

[12] Ramírez-Ayala, O., González-Hernández, I., Salazar, S., Flores, J., & Lozano, R. (2023). Real-Time Person Detection in Wooded Areas Using Thermal Images from an Aerial Perspective. *Sensors*, *23*(22), 9216. *https://doi.org/10.3390/s23229216*

[13] C. F. Silva and C. A. Siebra. An investigation on the use of convolutional neural network for image classification in embedded systems. in 2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Nov 2017, pp. 1–6L. Weiss, Instruction to Authors, Elsevier Publishing, *http://www.elsevier.com/authors.html, 1999, retrieved May 13, 2010*

[14] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al. Tensorflow: A system for large scale machine learning. in 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), 2016, pp. 265

[15] TensorFlow, Tensorflow: An end-to-end open source machine learning platform. 2016, *https://www.tensorflow.org/* (accessed May 10, 2019).

[16] Xiong, D., Long, Y., Yan, D., Lu, X., Ji, Z., & Fang, H. (2009). Surface morphology of soil cracks in Yuanmou Dry-hot Valley Region, Southwest China. *Journal of Mountain Science*, *6*(4), 373–379. *https://doi.org/10.1007/s11629-009-1059-6*

[17] Shi, B., Chen, S., Han, H., & Zheng, C. (2014). Expansive Soil Crack Depth under Cumulative Damage. *The Scientific World Journal*, *2014*, 1–9. *https://doi.org/10.1155/2014/498437*

[18] Li, G., Zhao, X., Du, K., Ru, F., and Zhang, Y. (2017). Recognition and evaluation of bridge cracks with modified active contour model and greedy search-based support vector machine: Automation in Construction, Vol. 78, pp. 51–61, *https://doi.org/10.1016/j.autcon.2017.01.019.*

[19] Dai, B., Gu, C., Zhao, E., Zhu, K., Cao, W., and Qin, X. (2019). Improved online sequential extreme learning machine for identifying crack behavior in the concrete dam: Advances in Structural Engineering, Vol. 22, No. 2, pp. 402–412, *https://doi.org/10.1177/1369433218788635.*

[20] Andrushia A, D., N, A., Lubloy, E., & G, P. A. (2021). Deep Learning Based Thermal Crack Detection on Structural Concrete Exposed to Elevated Temperature. *Advances in Structural Engineering*, *24*(9), 1896–1909. *https://doi.org/10.1177/1369433220986637*

[21] Choudhary, G.K., and Dey, S. (2012). Crack detection in Concrete Surfaces using Image Processing, Fuzzy Logic, and Neural Networks: 2012 IEEE 5th International Conference on Advanced Computational Intelligence, ICACI 2012, pp. 404–411, *https://doi.org/ 10.1109/ICACI.2012.6463195.*

[22] Luo, X., Wu, H., Yuan, H., & Zhou, M. (2020). Temporal Pattern-Aware QoS Prediction via Biased Non-Negative Latent Factorization of Tensors. *IEEE Transactions on Cybernetics*, *50*(5), 1798–1809. *https://doi.org/10.1109/TCYB.2019.2903736*

[23] Cha, Y., Choi, W., & Büyüköztürk, O. (2017). Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Computer-Aided Civil and Infrastructure Engineering*, *32*(5), 361–378. *https://doi.org/10.1111/mice.12263*

[24] Fan, Z., Li, C., Chen, Y., Wei, J., Loprencipe, G., Chen, X., & di Mascio, P. (2020). Automatic Crack Detection on Road Pavements Using Encoder-Decoder Architecture. *Materials*, *13*(13), 2960. *https://doi.org/10.3390/ma13132960*

[25] Ali, L., Alnajjar, F., Jassmi, H. al, Gocho, M., Khan, W., & Serhani, M. A. (2021). Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures. *Sensors*, *21*(5), 1688. *https://doi.org/10.3390/s21051688*

[26] Mathe, S. E., Pamarthy, A. C., Kondaveeti, H. K., & Vappangi, S. (2022). A Review on Raspberry Pi and its Robotic Applications. *2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP)*, 1–6. *https://doi.org/10.1109/AISP53593.2022.9760590*

[27] Vujovic, V., & Maksimovic, M. (2014). Raspberry Pi as a Wireless Sensor Node: Performances and Constraints. *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1013–1018. *https://doi.org/10.1109/MIPRO.2014.6859717*

[28] Kamath, R., Balachandra, M., & Prabhu, S. (2019). Raspberry Pi as Visual Sensor Nodes in Precision Agriculture: A Study. *IEEE Access*, *7*, 45110–45122. *https://doi.org/10.1109/ACCESS.2019.2908846*

[29] M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. *https://doi.org/10.1007/s11263-015-0816-y*

[30] Gawande, U., Hajari, K., & Golhar, Y. (2023). Real-Time Deep Learning Approach for Pedestrian Detection and Suspicious Activity Recognition. *Procedia Computer Science*, *218*, 2438–2447. *https://doi.org/10.1016/j.procs.2023.01.219*

[31] Nitin R. Gavai; Yashashree A. Jakhade; Seema A. Tribhuvan; Rashmi Bhattad MobileNets for Flower Classification using TensorFlow, International Conference on Big Data, IoT and Data Science(BID) *https://doi.org/ 10.1109/BID.2017.8336590, pg154 – 158, IEEE, 2017*

[32] Subhi, M. A., & Md. Ali, S. (2018). A Deep Convolutional Neural Network for Food Detection and Recognition. *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, 284–287. *https://doi.org/10.1109/IECBES.2018.8626720*