XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGENEERING

Application of Artificial Intelligence in River Pollution Monitoring to Support Environmental Management

Jullia Fernandes Felizardo[1], Thabatta Moreira Alves de Araújo[2]

[1]*Department of Computer Science, Federal Center of Technological Education of Minas Gerais*
Rua Álvares de Azevedo 400 - Bela Vista - Divinópolis - MG - Brasil - CEP 35503-822
julliacefet@gmail.com
Rua Álvares de Azevedo 400 - Bela Vista - Divinópolis - MG - Brasil - CEP 35503-822
thabatta@cefetmg.br

Abstract. Water pollution caused by visible solid waste has worsened in recent decades, compromising aquatic ecosystems, public health, and water security. Traditional monitoring methods present limitations in terms of spatial coverage and scalability. This work proposes a low-complexity model based on Convolutional Neural Networks (CNNs) for the automatic detection of visible human-eye-level waste in rivers from user-provided images. A labeled dataset was developed through the combination of different databases and manual annotation. Transfer Learning was applied using the MobileNetV2 architecture, chosen for its computational efficiency and low energy consumption, making it suitable for deployment on resource-constrained devices. Multiple independent training sessions were performed to ensure statistical robustness. The model achieved high and stable performance over 20 runs, with an average accuracy of 89.7%, an F1-score of 90.9%, and an area under the ROC curve (AUC) of 0.996, highlighting a strong discriminative capability. The absence of false negatives in the best-performing model reinforces its applicability in conservative environmental surveillance systems. The results indicate the feasibility, scalability, and potential of the proposed approach for automated river monitoring. Future integration with geospatial platforms and mobile devices is suggested to expand its adoption by environmental agencies and public managers.

Keywords: CNN; pollution detection; binary classification; water monitoring; environmental surveillance

1 Introduction

The degradation of water resources, driven by population growth and unplanned urbanization, is a global environmental concern. The river pollution poses serious risks to biodiversity, water security, and public health [1]. However, addressing these risks is hindered by the limitations of traditional monitoring methods, which rely on in-person inspections and laboratory analysis and are often costly, time-consuming, and spatially limited [2].In this context, the Artificial Intelligence (AI) approaches, especially image classification and pattern detection, have shown great potential for environmental monitoring. In this way, the Convolutional Neural Networks (CNNs) stand out due to their ability to extract hierarchical visual features [3, 4].

Although CNNs achieve excellent results, the most popular models like ResNet and DenseNet often require high-performance hardware, such as GPUs, which may not be available in low-resource environments, common in decentralized environmental monitoring initiatives [5]. Given this challenge, lightweight models have emerged as viable alternatives that combine satisfactory performance with minimal computational and energy requirements, rendering them particularly suitable for mobile and embedded systems, frequently employed in remote monitoring [5]. Therefore, this study seeks to investigate the following research question: Are lightweight models effective for environmental monitoring, specifically in the detection of visible pollution in river imagery?

Based on the question, this study aims to evaluate the applicability of a CNN-based approach for environmental monitoring using a low complexity model, MobileNetV2. To this end, a labeled dataset was constructed with river images classified as “polluted” and “non-polluted,” and a Transfer Learning. The main objectives of the study are to (i) develop a binary-labeled image dataset, (ii) train a CNN model for binary classification, and (iii) assess the model’s performance using standard classification metrics.

This article is structured into four main sections. The Background Concepts section summarizes key terms and theoretical foundations. The Model Architecture and Experimental Setup section outlines the methodology for building the classifier, including data preparation. The Results and Discussion section presents evaluation

metrics and analyzes the outcomes. Finally, the Conclusion summarizes the findings and suggests future research directions.

2 Background Concepts

The acquisition of data without physical contact is crucial for monitoring large areas in diverse scenarios, such as environmental, engineering structures, among others [6]. This remote monitoring is possible due to advanced devices in sensors, telecommunications networks, and strategies for extracting learning from the data acquisition and revealing patterns to support human decision-making. Some techniques of Machine Learning based on Artificial Intelligence (AI) are employed efficiently in these strategies [4].

In image processing, Convolutional Neural Networks (CNNs) excel at computer vision through hierarchical feature extraction [3, 7]. While classic CNNs often demand advanced computational processing resources, lightweight alternatives like MobileNetV2 (**Figure 1**) feature a more compact architecture, thus requiring fewer resources. They employ depthwise separable convolutions, factoring standard convolutions into depthwise and pointwise steps, to reduce computations by approximately 8x [5, 8]. Specifically, MobileNetV2 uses inverted residual blocks to maintain high accuracy at a lower computational cost compared to heavier architectures like ResNet and VGG, making it ideal for resource-constrained devices [5, 7].

Moreover, the Transfer Learning approach is used in image processing by employing CNN models, which are pre-trained on large datasets to recognize visual patterns. By reusing learned weights, these models can be quickly adapted to new image classification tasks, making it a popular strategy for limited datasets. [9].

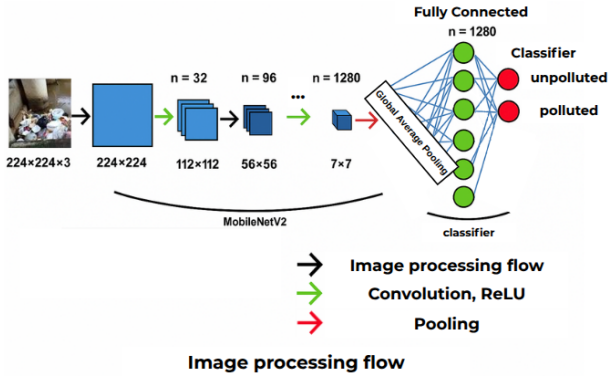


Figure 1. MobileNetV2 network architecture

3 Model Architecture and Experimental Setup

This section describes the methodology used for developing the river waste detection model, following the general workflow illustrated in **Figure 2**. All development was carried out on a Dell notebook (Intel Core i7, 16 GB RAM, Intel Iris Xe Graphics, Ubuntu 22.04 LTS) using Python 3.11 and Jupyter Notebooks. The main libraries used were TensorFlow/Keras for Deep Learning, NumPy for numerical operations, OpenCV and Pillow for image manipulation, Scikit-learn for evaluation metrics, and Matplotlib/Seaborn for visualization.

3.1 Data Collection and Preprocessing

The process began with **Data Acquisition and Organization (Steps 1–3 in Figure 2)**. The dataset was built with river images, classified as either 'polluted' (litter visible to the human eye) or 'non-polluted' (without visible waste). Images were manually collected and labeled from platforms with Creative Commons licenses and supplemented by a dataset from the **Roboflow** platform (totaling 1,887 images for training, 116 for validation, and 61 for testing, before data augmentation).

Next, the **Image Preprocessing (Steps 4–6)** included resizing all images to **224x224 pixels**, converting them to RGB format, and organizing the folder structure for efficient data reading.

To mitigate overfitting and expand the training set, **Data Augmentation (Steps 7–9)** was applied. Using the Keras library, two data generators were defined: **datagen_train**, with intense transformations (zoom, rotation, shifts, flipping, and brightness adjustments) applied to the training set; and **datagen_val_test**, with lighter transformations (pixel normalization, small brightness variations, and horizontal flipping) for the validation and test sets.

A class-specific augmentation strategy was employed for balancing: the ”polluted” class had its training images replicated 2 times, while the ”non-polluted” class had 9 replicas, in addition to one augmented copy with minor transformations in the validation and test sets.

The subsets were consolidated with the Roboflow data, totaling **3,730** ”non-polluted” and **3,458** ”polluted” images for **training**; **160** ”non-polluted” and **170** ”polluted” for **validation**; and **160** ”non-polluted” and **171** ”polluted” for **testing**. All data were organized for reading via *flow_from_directory* in batches of 16 samples.

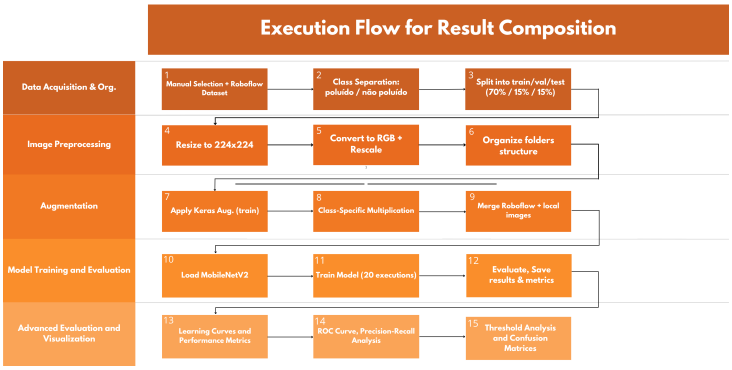


Figure 2. General workflow of the model development process.

3.2 Architecture, Training, and Evaluation

The **Model Development and Training (Steps 10–12)** were based on Transfer Learning, using the MobileNetV2 architecture pre-trained on ImageNet. The detailed structure of the model is presented in **Figure 3**. The base layers of MobileNetV2 (including the initial convolutional blocks and the Inverted Residual Blocks 1 to 17) were frozen to preserve the learned features.

On top of this base, custom layers were added: a Global Average Pooling layer to reduce dimensionality, followed by two Dropout layers (30% rate) to mitigate overfitting. Between the Dropouts, a dense layer with 64 neurons and ReLU activation was inserted. The output layer contains a single neuron with Sigmoid activation for binary classification (”polluted”/”non-polluted”). The model was compiled using the Adam optimizer (learning rate of 0.0001), binary cross-entropy loss function, and accuracy as the performance metric.

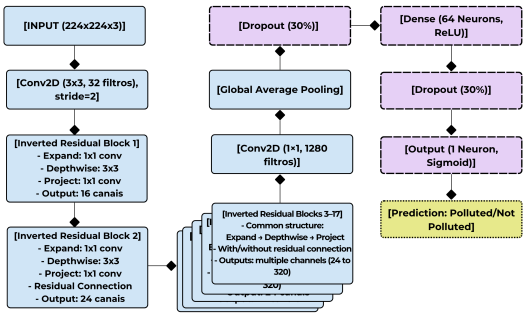


Figure 3. Process Workflow Design.

The training was run independently 20 times, each for 20 epochs, with validation on a separate dataset. To address class imbalance, class weights inversely proportional to sample frequency were applied. Accuracy and loss histories were recorded. The model with the best F1-Score on the test set was saved in `.h5` format. A threshold of 0.5 was used to classify predictions as ”polluted” or ”non-polluted”.

For **Performance Evaluation (Steps 13–15)**, the model was evaluated on the test set using the following metrics: Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and Classification Report. Learning curves, ROC curves, and Precision-Recall curves were also analyzed. Additionally, an analysis of the impact of different thresholds on classification was performed. Detailed results from the 20 runs, including means and standard deviations, were stored for statistical analysis. The full source code is available at [10].

4 Results and Discussion

The proposed model demonstrated stable performance across 20 runs, exhibiting low variability among the results. Test accuracy consistently remained close to 90%, and a standard deviation of only 2.1% reinforced the consistency of the performance. **Table 1** details that the average F1-score on the test set was 90.96%, indicating an excellent generalization capability. Visually, the boxplots **(Figure 4)** illustrate this stability, with reduced dispersions and the absence of significant outliers, confirming the reliability of the obtained metrics.

The learning curves, presented in **Figure 5**, suggest stable and efficient convergence. A coordinated growth of training and validation accuracy was observed, while the loss continuously decreased. Although the validation loss (0.2918) was slightly higher than the training loss (0.2132), no indications of overfitting were identified, reinforcing the model's robustness and its ability to generalize to unseen data.

Table 1. Averages and Standard Deviations of Metrics (20 Runs).

Metric	**Average**	**Standard Deviation**
Train Accuracy	0.9153	0.0027
Train Loss	0.2132	0.0039
Val Accuracy	0.8692	0.0169
Val Loss	0.2918	0.0347
Test Accuracy	0.8970	0.0213
F1 on Test	0.9096	0.0169

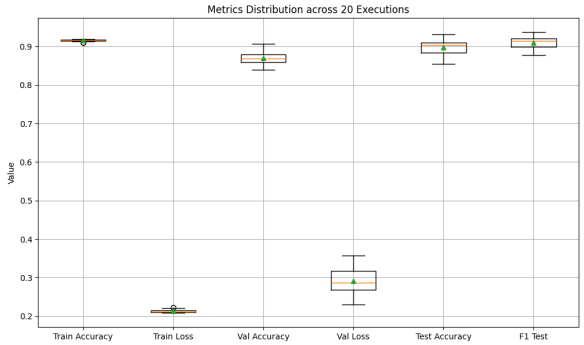


Figure 4. Distribution of metrics across 20 runs.

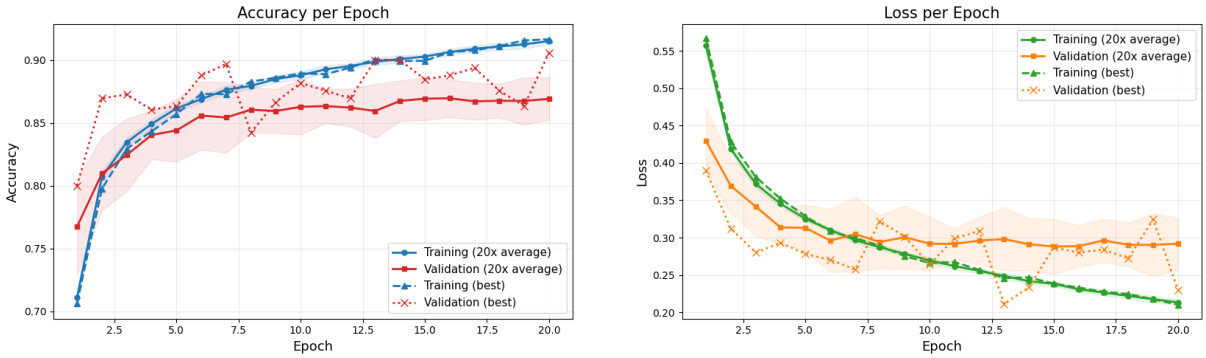


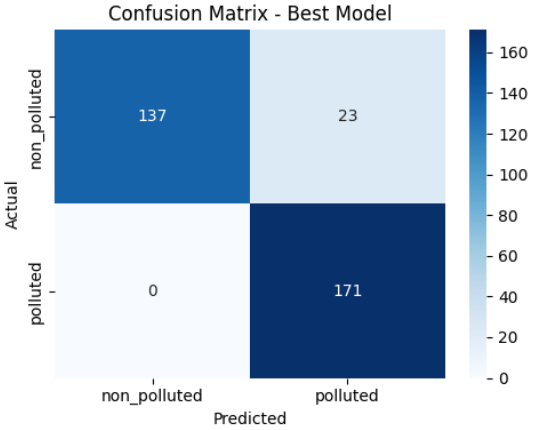
Figure 5. Model Convergence Curves: Accuracy and Loss per Epoch (average of 20 runs and best model).

Moving to a detailed analysis of the best model on the test set, the Confusion Matrix **(Figure 6a)** demonstrates the classifier's high performance, which correctly categorized all 171 images labeled as "polluted". Notably, only 23 false positives were recorded among the 160 'unpolluted' images, a crucial result for environmental surveillance systems, as it establishes zero false negatives for the polluted class. This implies that the vast majority of images

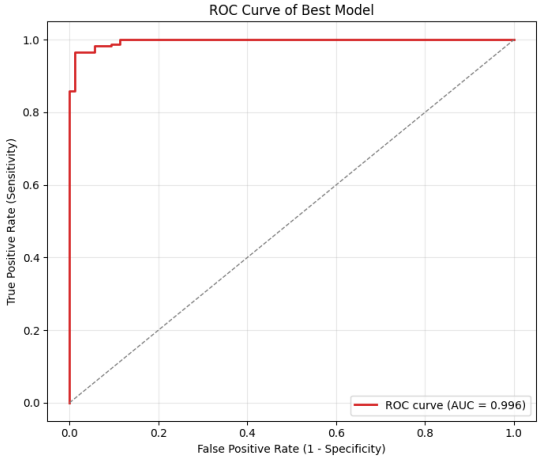
classified as "polluted" truly belong to this category, reflecting a high precision of 0.88 and a recall of 1.00 for the class of interest.

The ROC and Precision-Recall curves, displayed in **Figures 6b** and **6c**, respectively, corroborate the model's exceptional discriminative capability. With an AUC of 0.996, the model demonstrates high separability between classes. The AP (Average Precision) value of 0.996 is equally significant, indicating that the model maintains high precision even at high recall levels, a fundamental characteristic in sensitive environmental contexts, where failure to detect an event can have severe consequences.

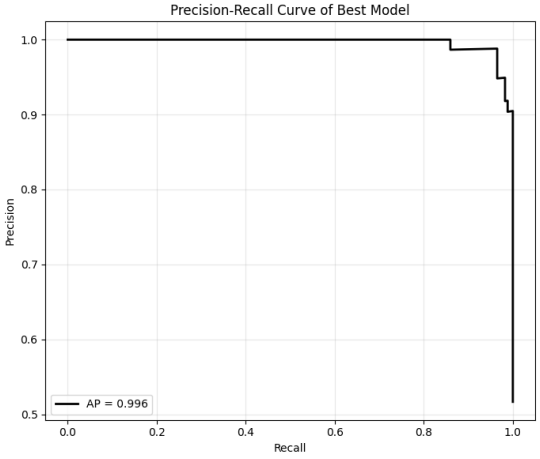
This strong performance is further reflected in **Table 2**, which consolidates the detailed classification metrics of the best model and reinforces the precision-recall balance previously discussed.



(a) Confusion Matrix of the Best Model.



(b) ROC Curve of the Best Model.



(c) Precision-Recall Curve of the Best Model.

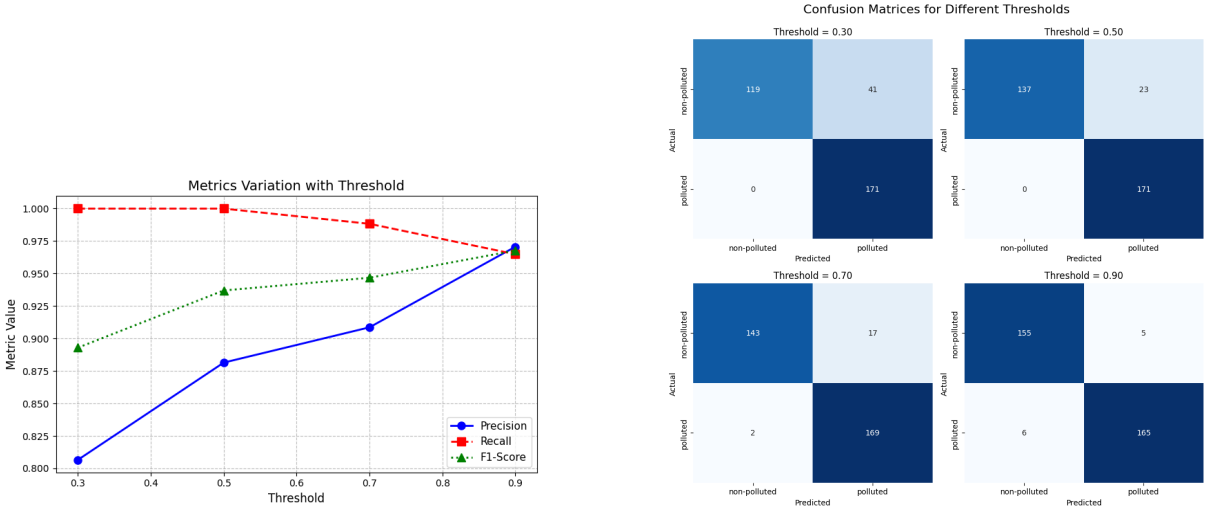
Figure 6. Performance metrics of the best classification model.

Table 2. Classification Report of the Best Model.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
unpolluted	1.00	0.86	0.92	160
polluted	0.88	1.00	0.94	171
accuracy		0.93		
macro avg	0.94	0.93	0.93	331
weighted avg	0.94	0.93	0.93	331

The investigation into the impact of the classification threshold, illustrated in **Figures 7a** and **7b**, and consol-

idated in **Table 3**, highlights the importance of its choice on performance. Lower thresholds (e.g., 0.30) guarantee full recall, albeit with an increase in false positives. Conversely, higher thresholds (e.g., 0.90) reduce false positives but may introduce false negatives. A threshold of 0.50 was identified as the most balanced point, ensuring maximum recall (1.00), an F1-score of 0.937, and a manageable number of false positives. This configuration makes it ideal for applications where not missing critical events is more important than avoiding false alarms, such as in environmental monitoring.



(a) Metric Variation vs. Threshold.

(b) Confusion Matrix of Different Thresholds.

Figure 7. Analysis of classification thresholds.

Table 3. Comparison of metrics for different thresholds.

Threshold	Precision	Recall	F1-Score	FP	FN
0.30	0.807	1.000	0.893	41	0
0.50	0.881	1.000	0.937	23	0
0.70	0.909	0.988	0.947	17	2
0.90	0.971	0.965	0.968	5	6

Finally, the integrated analysis of the results suggests the consistency, sensitivity, and practical applicability of the proposed approach for automated environmental monitoring. The stability demonstrated across multiple runs and its high discriminative power solidify the classifier as a reliable tool. The selection of a threshold of 0.50, which ensures **maximum recall and the elimination of false negatives**, is particularly relevant for scenarios where detection failures can lead to severe consequences. This study, therefore, establishes the technical feasibility of the proposed solution for automated water quality surveillance.

5 Conclusions

This work developed and evaluated an intelligent system for the automatic detection of visible pollution in rivers using images and CNNs. The model, based on Transfer Learning with the MobileNetV2 architecture and trained on a robust and augmented dataset, proved to be effective in the binary classification of rivers as either ”polluted” or ”non-polluted”. The results confirm the model’s high performance and consistency, achieving an average accuracy of 89.7% and an F1-score of 90.9% on the test set. The ROC curves (AUC of 0.996) and Precision-Recall curves (AP of 0.996) demonstrate their excellent ability to discriminate between classes, with emphasis on the absence of false negatives in the best run, which is crucial for environmental alert systems.

While the augmented dataset from Roboflow enhanced model robustness through diverse scenarios and conditions, future research should validate performance on non-augmented, geographically diverse datasets to ensure generalizability across different water bodies and pollution types. Additionally, investigating domain adaptation

techniques and collecting data from various river basins worldwide would strengthen the model's real-world applicability.

From a practical standpoint, MobileNetV2's computational efficiency makes the model suitable for deployment on resource-constrained devices, such as smartphones and drones, allowing for real-time and large-scale inference even in regions with limited infrastructure. To demonstrate its applicability, a prototype interface was developed to facilitate the use of the model on various devices. Through this interface, users can upload an image, and its metadata (such as capture date and location) is processed to immediately return the classification result regarding the presence of pollution. More details and access to this prototype can be found at: [11]. An inherent limitation is the focus on visible pollution detection, excluding submerged pollutants or differentiation by type.

As a contribution, this study offers a low-cost, scalable, and replicable solution for water monitoring, supporting environmental management. Future research may explore integration with Geographic Information Systems (GIS), expansion of the dataset to various river basins, and the use of multimodal approaches (image + metadata) to further improve accuracy.

Acknowledgements. We thank the Federal Center for Technological Education school community of Minas Gerais (CEFET-MG), the Divinópolis campus, and the Directorate of Research and Graduate Studies (DPPG) of CEFET-MG. The authors gratefully acknowledge the financial support from FAPEMIG, which made it possible to present this work at the congress.

Authorship statement. I confirm that I am solely responsible for the content and authorship of this work, and that all material included is either my own or used with permission.

References

[1] R. Kumar, M. Qureshi, D. K. Vishwakarma, N. Al-Ansari, A. Kuriqi, A. Elbeltagi, and A. Saraswat. A review on emerging water contaminants and the application of sustainable removal technologies. *Case Studies in Chemical and Environmental Engineering*, vol. 6, pp. 100219, 2022.

[2] P. SRIVASTAVA, Y. MITTAL, S. GUPTA, R. ABBASSI, and V. GARANIYA. Recent progress in biosensors for wastewater monitoring and surveillance. In M. ASADNIA, A. RAZMJOU, and A. BEHESHTI, eds, *Artificial Intelligence and Data Science in Environmental Sensing*, Cognitive Data Science in Sustainable Computing, pp. 245–267. Academic Press, Londres, 1 edition. Capítulo 11, 2022.

[3] I. S. SILVA. Inteligência artificial para avaliação da qualidade da água. Dissertação (mestrado em recursos hídricos), Universidade Federal de Sergipe, São Cristóvão, SE. Orientador: Carlos Alexandre Borges Garcia, 2019.

[4] P. H. B. d. BARROS and A. M. d. FREITAS JUNIOR. Combinando inteligência artificial e imagens de satélite para a previsão de sinistros agrícolas: Uma nota. *Revista Brasileira de Economia*, vol. 77, pp. e012023, 2023.

[5] de T. M. A. Araujo, de C. A. Mattos Teixeira, and C. R. L. Francês. Enhancing geotechnical damage detection with deep learning: a convolutional neural network approach. *PeerJ Computer Science*, vol. 10, pp. e2052, 2024.

[6] R. A. Schowengerdt. Chapter 1 - the nature of remote sensing. In R. A. Schowengerdt, ed, *Remote Sensing (Third Edition)*, pp. 1–X. Academic Press, Burlington, third edition edition, 2007.

[7] M. Akay, Y. Du, C. Sershen, M. Wu, T. Chen, S. Assassi, and Y. Akay. Deep learning classification of systemic sclerosis skin using the mobilenetv2 model. *IEEE Open Journal of Engineering in Medicine and Biology*, vol. PP, pp. 1–1, 2021.

[8] J. G. G. Silva, T. M. Araújo, E. H. B. Maia, and C. R. L. Francês. Moisture detector in concrete using convolutional neural networks. In *Proceedings of the XLV Ibero-Latin-American Congress on Computational Methods in Engineering, ABMEC*, Maceió, Alagoas, Brazil, 2024.

[9] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, vol. 27, 2014.

[10] J. Fernandes. Binary classification of a polluted river - github repository, 2025a.

[11] J. Fernandes. Classificador binário de poluição em rios. Disponível em: `https://huggingface.co/spaces/JulliaFernandes/Classificador_Binario_de_Poluicao_em_Rios`. Acesso em: jul. 2025, 2025b.