Automated Leak Classification in Drip Irrigation Systems using Deep Learning and RGB Cameras

Fernando Pujaico Rivera[1], Carlos Kamienski[1]

1 Universidade Federal do ABC, SP, Brazil

A paper from the Proceedings of the

17th International Conference on Precision Agriculture and

11th Brazilian Congress on Precision and Digital Agriculture

13-15 July 2026

Porto Alegre, Brazil

Abstract.

The increasing demand for water efficiency in agriculture has driven the development of intelligent irrigation systems. Among them, drip irrigation is widely adopted due to its efficiency; however, these systems are susceptible to leaks caused by mechanical wear, animal interference, and adverse environmental conditions. Manual inspection of drip irrigation systems is time-consuming, difficult to scale, and prone to human error, motivating the need for automated solutions based on computer vision and deep learning. This work addresses the problem of automatic leak classification in drip irrigation systems using RGB video data. We propose a lightweight deep learning pipeline composed of dataset acquisition in real field conditions, video processing and annotation, and model training using compact CNN-LSTM architectures designed for edge deployment. The system operates under natural illumination variability and is robust to visual noise such as soil background, vegetation, water reflections, and viewpoint changes. The proposed approach achieves an accuracy of 0.9252 and an F1-score of 0.9107 in the binary leak classification scenario. The main contribution of this work is to provide evidence of the feasibility of lightweight RGB-only deep learning solutions for leak classification in real agricultural environments, suggesting that reliable leak classification can be achieved without specialized sensors.

Keywords.

Drip Irrigation Systems, Leak classification, Deep learning, Convolutional Neural Networks (CNNs)

1. Introduction

Drip irrigation is widely used in modern agriculture due to its high water-use efficiency. However, irrigation lines are vulnerable to leaks caused by component degradation, animal activity, and environmental conditions. In practice, leak inspection is still largely performed manually, making large-scale monitoring difficult and time-consuming.

Several approaches for irrigation leak detection have been proposed in the literature. Previous works explored thermal cameras, multispectral imagery, UAV platforms, and alternative sensing strategies such as vibration-based monitoring [8, 5, 3, 7, 2, 1, 4]. Although effective in specific scenarios, many of these solutions depend on specialized sensors or costly acquisition platforms, limiting their applicability in resource-constrained agricultural environments.

In this work, we propose an RGB-based deep learning approach for leak classification in drip irrigation systems using video data collected under real field conditions. The proposed pipeline combines lightweight CNN-LSTM architectures with temporal video modeling, targeting deployment on edge-computing devices.

This work investigates the feasibility of RGB-only leak classification under realistic agricultural conditions and evaluates compact architectures suitable for edge-oriented deployment.

2. Methodology

The proposed methodology evaluates two CNN-LSTM video classification architectures (Section 2.2) using a field-collected dataset (Section 2.1). The dataset is divided into training/validation and test subsets, and the models are compared through 5-fold cross-validation (Section 2.3) followed by statistical analysis of the validation results (Section 2.4). The best-performing architecture is then evaluated on the test set (Section 2.5).

2.1 Dataset

2.1.1 LCDIS Dataset

The Leak Classification in Drip Irrigation Systems (LCDIS) dataset was collected at the experimental farm operated by EMBRAPA (Brotas-SP, Brazil). Videos were captured along irrigation lines under realistic agricultural settings using various RGB cameras, such as the GoPro Hero 12 Black, iPhone 15, Oscal S60 Pro, and Redmi Note 11, to enhance diversity in recording conditions and sensor characteristics.

Leak incidents were deliberately introduced to guarantee the occurrence of various leakage types. Spray-type leaks were produced by making small holes in the irrigation lines with a blade. After partially closing these holes, imperfect repairs resulted in excessive dripping situations. Additional samples representing normal conditions were obtained when the irrigation system operated without any visible irregularities. Table 1 presents statistics on the length of the collected videos.

2.1.2 Annotation Protocol

Videos were manually annotated into five categories representing different irrigation conditions or labels: other (0), corresponding to unrelated or non-informative scenes; with-hose (1), representing hose segments without visible dripping; with-leak-drip (2), corresponding to abnormal dripping leaks; with-leak-spray (3), representing spray-type leaks; and without-leak (4), corresponding to normal irrigation dripping.

Although the dataset is annotated using five classes, the final objective is binary leak classification. Therefore, the five-class structure is used as an intermediate representation of irrigation conditions before remapping into two categories: leak, composed of the classes with-leak-drip and with-leak-spray; and no-leak, composed of without-leak, with-hose, and other.

Table 1. Video duration statistics. The first column lists the classification categories within the video dataset. The second column indicates the number of videos for each category. The third and fourth columns show the average and standard deviation. The last column presents the total cumulative duration of the videos. All time measurements are given in seconds.

Class	N	Mean (s)	Std (s)	Total (s)
total	185	14.55	41.18	2690.97
other	15	108.08	95.70	1621.22
with-hose	12	34.49	52.24	413.89
with-leak-drip	19	4.71	2.51	89.49
with-leak-spray	40	4.69	2.16	187.52
without-leak	99	3.83	1.86	378.86

The five-category annotation scheme was maintained during training in order to preserve finer-grained supervision signals associated with different irrigation conditions, while the final evaluation focuses on the practical binary leak classification task.

2.1.3 Dataset Splitting

The dataset was divided using stratified sampling, resulting in 154 videos for cross-validation and 31 videos for testing.

2.2 Classification Architecture

The proposed video classification pipeline follows a standard CNN-LSTM design, where convolutional neural networks extract spatial features from individual frames and a recurrent module models temporal dependencies across sequences.

Two backbone architectures implemented in the Torchvision framework were evaluated: MobileNetV3-Small and EfficientNet-B0. These models differ in capacity and computational cost, enabling analysis of the trade-off between efficiency and accuracy.

All configurations follow the same processing pipeline: frame-level feature extraction using a CNN backbone, temporal aggregation using a bidirectional LSTM, and final classification through a fully connected layer. A summary of the architectural components is provided in Table 2.

Table 2. Simplified CNN-LSTM pipeline used for video classification.

Stage	Description
Input	Video sequence (16 frames)
Frame Encoder	CNN backbone (MobileNetV3-Small / EfficientNet-B0)
Feature Extraction	Global average pooling per frame
Temporal Modeling	Bidirectional LSTM (2 layers, 256 hidden units)
Temporal Aggregation	Last timestep output
Regularization	Dropout (p = 0.4)
Classifier	Fully connected layer (logits output)

Model parameter statistics are presented in Table3, and all models were trained using the configuration described in Section 2.3.2.

Table 3. Model parameter comparison. "Feat. Dim." denotes the feature dimensionality produced by the backbone before being fed into the LSTM. "Backbone (M)" indicates the number of parameters (in millions) of the convolutional feature extractor alone. "LSTM (M)" corresponds to the parameters of the temporal modeling module. "Classifier (M)" represents

the parameters of the final fully connected layer. Finally, "Total (M)" reports the overall number of parameters of the complete CNN-LSTM architecture.

Backbone	Feat. Dim.	Backbone (M)	LSTM (M)	Classifier (M)	Total (M)
mobilenet_v3_small	576	0.927	3.285	0.0026	4.215
efficientnet_b0	1280	4.008	4.727	0.0026	8.737

2.3 Cross-validation Process

To evaluate the classification architectures under different training conditions, a K-fold cross-validation strategy with $K = 5$ was employed. The evaluated models correspond to different CNN-LSTM configurations using distinct backbone architectures. The training and validation dataset was partitioned into five disjoint subsets. In each fold, one subset was used for validation while the remaining four were used for training, resulting in five independently trained models and corresponding evaluation metrics. A stochastic temporal sampling strategy was applied during training. Details of the data variability strategy and training configuration for each fold are provided in Sections 2.3.1 and 2.3.2, respectively. The code used to train the models and perform cross-validation is publicly available in the repository [6].

2.3.1 Temporal Sampling Strategy

A stochastic temporal sampling strategy was applied during training, validation, and testing in order to increase sequence diversity and obtain a more robust performance estimate. Instead of fixed frame positions, each video sequence is generated by randomly selecting the starting time ($t0$) and the temporal sampling rate (fps).

The sequence length remains fixed at $L = 16$ frames, while the sampling rate is randomly selected within a predefined range [7, 9] fps. This allows the model to observe the same video under different temporal resolutions while preserving a consistent input size. When videos are shorter than the required temporal window, frames are reused cyclically to maintain a fixed-length sequence.

No spatial augmentation techniques were applied. Therefore, temporal stochastic sampling is the only source of data variability used during training.

2.3.2 Training Fold

The CNN backbones were initialized with pretrained ImageNet weights and kept frozen, while only the LSTM and classification layers were trained using cross-entropy loss.

Training was performed for up to 100 epochs using the Adam optimizer with a learning rate of $10 - 4$ and batch size of 2. Early stopping with a patience of 25 epochs was applied based on the mean validation loss, and all experiments used a fixed random seed of 42. Input frames were resized to 512×512 and normalized to the range [0, 1].

Validation was repeated three times per epoch using different temporal samples, and the mean validation loss was used for early stopping. The model with the lowest validation loss was selected for testing at the end of each fold.

2.4 Statistical Comparison of Architectures

A one-sided Wilcoxon signed-rank test was used to compare architectures based on validation accuracies across the $K = 5$ cross-validation folds. The test was chosen due to the paired nature of the samples and its non-parametric assumption-free formulation.

The null hypothesis states that there is no significant difference between architectures A and B, while the alternative assumes superior performance of A. Statistical significance was defined as $p < 0.05$.

2.5 Testing Process

After the 5-fold cross-validation procedure, the best model from each fold was selected for testing. Each model evaluated the complete test set 10 times using stochastic temporal sampling, producing different frame sequences at each repetition. Predictions obtained from all repetitions and folds were aggregated for evaluation. The repeated evaluations were used to estimate prediction robustness under stochastic temporal sampling rather than to artificially increase the effective dataset size.

Performance was evaluated using precision, recall, F1-score, accuracy, and confusion matrices in the binary leak classification setting. The original five dataset categories were remapped into leak and no-leak classes for final evaluation.

3 Results

3.1 Cross-validation Results

The validation results obtained during the 5-fold cross-validation experiments are presented in Table 4. These statistics summarize both the central tendency and variability of each architecture, which is referenced by the name of its backbone.

Table 4. Descriptive statistics per architecture in validation accuracy. The Mean column represents the average validation accuracy across folds, the Std column indicates the corresponding standard deviation, and the CI (95%) column reports the confidence interval of the mean in the form [lower, upper].

Architecture (backbone)	Mean	Std	CI (95%)
mobilenet_v3_small	0.7794	0.0631	[0.7011, 0.8577]
efficientnet_b0	0.8164	0.0394	[0.7675, 0.8653]

3.2 Statistical Tests on Cross-validation Results

A one-sided Wilcoxon signed-rank test was applied to the validation accuracies across the 5 cross-validation folds (Table 4) to evaluate whether performance differences between architectures were statistically significant.

No significant difference was found between EfficientNet-B0 and MobileNetV3-Small ($p > 0.05$). The hypothesis that EfficientNet-B0 outperforms MobileNetV3-Small yielded $p = 0.2188$, while the reverse comparison resulted in $p = 0.8438$.

3.3 Test Results

Based on the cross-validation results, the EfficientNet-B0 architecture was selected for final evaluation on the test set, as the Wilcoxon test did not indicate a statistically significant difference between models; therefore, the choice was based solely on the mean validation performance. Although the models were trained using the original five-category annotation scheme, the final evaluation focuses on the practical binary leak classification scenario. Table 5 presents the classification performance obtained on the test set for the binary classification task. Figure 1 presents the confusion matrix for the binary classification scenario.

Table 5. Classification report on the test set for the binary leak classification task. Precision, recall, F1-score, and support are reported for each class. A total of 1550 aggregated evaluations were performed (31 videos × 5 models × 10 repetitions).

Class	Precision	Recall	F1-score	Support
leak	0.9507	0.8100	0.8747	500
no-leak	0.9155	0.9800	0.9466	1050
Macro avg	0.9331	0.8950	0.9107	1550

Weighted avg	0.9268	0.9252	0.9234	1550
Accuracy			0.9252	1550

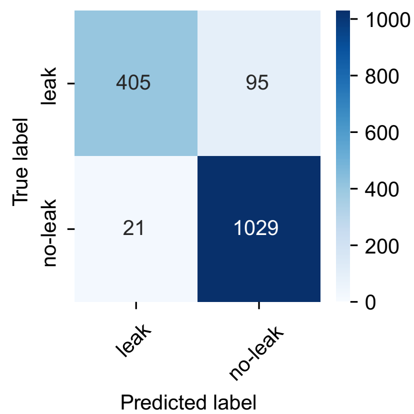

Fig 1. Confusion matrix for the binary leak classification task using the EfficientNet-B0 backbone. Rows represent ground truth labels and columns correspond to predicted labels.

4 Discussion

4.1 Architecture Comparison

Both architectures achieved comparable validation performance, with EfficientNet-B0 showing a slight but non-significant advantage in accuracy and F1-score. This highlights a trade-off between performance and efficiency, where EfficientNet-B0 provides marginally better predictive performance, while MobileNetV3-Small is a more lightweight alternative suitable for resource-constrained deployment scenarios.

4.2 Binary Leak Classification Performance

The proposed approach achieved strong performance in the binary leak classification scenario (accuracy 0.9252 and F1-score 0.9107), suggesting that RGB-based video analysis with lightweight CNN-LSTM architectures can effectively distinguish leak and non-leak conditions in real agricultural environments without requiring specialized sensing hardware.

4.3 Dataset Limitations and Deployment Considerations

The dataset was collected under real-world field conditions with high variability in illumination, vegetation, soil, and viewpoints, making the classification task more challenging. Its relatively small size and class imbalance, as seen in Table 1, may have impacted the learning process, but it still provided sufficient diversity for the models to learn meaningful patterns.

4.4 Limitations and Future Work

This work presents several limitations, including limited dataset size, class imbalance, and the use of single-annotator labels. In addition, only temporal augmentation was applied and the CNN backbones remained frozen during training. Future work should focus on expanding the dataset, improving annotation reliability, investigating spatial augmentation and imbalance mitigation strategies, and evaluating inference performance in embedded robotic platforms under long-term field conditions.

6 Conclusions

This work investigated the feasibility of RGB-based deep learning for leak classification in drip irrigation systems using lightweight CNN-LSTM architectures under real-world agricultural conditions.

The proposed approach achieved an accuracy of 0.9252 and an F1-score of 0.9107 in the binary classification scenario, indicating that reliable leak detection can be performed using only RGB video data, without the need for specialized sensing hardware.

The main contribution of this study is the proposed CNN-LSTM pipeline and the empirical evidence that leak identification is feasible using only video-based information under the considered hardware and software constraints, supporting compact and practical solutions for real-world agricultural deployment.

Acknowledgments

This work was partially funded by the São Paulo Research Foundation (FAPESP) under grant 2024/15527-9 (SMART project) and by the INCT of Intelligent Communications Networks and the Internet of Things (ICoNIoT) funded by CNPq under grant 405940/2022-0 and CAPES under grant 88887.954253/2024-00..

References

[1] Jiawei Chen et al. “Augmenting a deep-learning algorithm with canal inspection knowledge for reliable water leak detection from multispectral satellite images”. In: Advanced Engineering Informatics 46 (2020), p. 101161. issn: 1474-0346. doi: 10.1016/j.aei.2020.101161.

[2] Yanbo Huang et al. “Airborne remote sensing for detection of irrigation canal leakage”. In: Irrigation and Drainage 59.5 (2010), pp. 524–534. doi: https://doi.org/10.1002/ird.511.

[3] Jean-Claude Krapez et al. “Multispectral Optical Remote Sensing for Water-Leak Detection”. In: Sensors 22.3 (2022). issn: 1424-8220. doi: 10.3390/s22031057.

[4] Nayna Ann Moni et al. “Water Leak Detection from Irrigation Pipelines in Botswana using Vibration Interpretation Technique”. In: 2019 IEEE AFRICON. 2019, pp. 1–6. doi: 10.1109/AFRICON46755.2019.9133829.

[5] Luís Pádua et al. “Detection of Leak Areas in Vineyard Irrigation Systems Using UAV-Based Data”. In: Drones 8.5 (2024). issn: 2504-446X. doi: 10.3390/drones8050187.

[6] Fernando Pujaico Rivera. Irrigation Leak Classification using Computer Vision. https://gitlab.com/precision-agriculture-ml/IrrigationLeakClassificationCV. Accessed: 2026-05-06. 2026.

[7] Seydou Traore et al. “Unmanned aerial vehicle system integrating high-resolution sensors for detecting leaks in irrigation canals–Proof of concept”. In: Irrigation and Drainage 71.1 (2022), pp. 206–217. doi: 10.1002/ird.2641.

[8] Levent Türkler, Taner Akkan, and Lütfiye Özlem Akkan. “Detection of Water Leakage in Drip Irrigation Systems Using Infrared Technique in Smart Agricultural Robots”. In: Sensors 23.22 (2023). issn: 1424-8220. doi: 10.3390/s23229244.