Incremental Learning Approaches for Flood Detection in Dynamic River Environments

Otávio Ferracioli Coletti[1], Saulo Matos Neves[1], Caetano Mazzoni Ranieri[2], Gowri Sankar Ramachandran[3], Raja Jurdak[3], Jó Ueyama[1]

[1] Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (USP)
São Carlos, SP, Brazil

[2]Instituto de Geociências e Ciências Exatas, Universidade Estadual Paulista (UNESP)
Rio Claro, SP, Brazil

[3]School of Computer Science, Queensland University of Technology (QUT)
Brisbane, QLD, Australia

{otaviocoletti,saulo.matos}@usp.br, joueyama@icmc.usp.br

cm.ranieri@unesp.br

{g.ramachandran,r.jurdak}@qut.edu.au

Abstract. *Identifying urban floods promptly can mitigate risks to public safety. Traditional models for this task rely on static datasets and are ill-suited to handle continuous environmental variation. This work presents an image-based flood monitoring system that integrates an incremental learning pipeline capable of adapting in real-time to distributional shifts. The model is periodically updated through cloud-based retraining informed by expert feedback. We evaluate different replay buffer strategies and demonstrate that the buffer replacement policy has a greater influence on performance than the buffer size. Our results show that incremental learning with selective replay significantly improves model robustness in dynamic flood monitoring scenarios.*

1. Introduction

Urban floods pose increasing risks to communities, especially under the effects of climate change. In Brazil, cities like São Carlos frequently experience such events [Morelli and Cunha 2021], with a record 263.6 mm rainfall in December 2022. More recently, in April–May 2024, severe flooding in Rio Grande do Sul impacted 441 cities, displaced over 600,000 people, and resulted in 179 deaths [Machado 2024].

Traditional flood monitoring approaches rely on static datasets and full model retraining whenever environmental conditions change. This process is costly, requires storing all past data, and delays deployment [Tian et al. 2024], which is impractical in real-world, high-frequency monitoring scenarios.

Rivers are dynamic environments, and data drift caused by changes in vegetation, lighting, infrastructure, or camera positioning can degrade model performance [Bayram et al. 2022]. While incremental updates can help, they risk **catastrophic forgetting**. To address this, we adopt replay-buffer strategies in our incremental learning pipeline to retain past knowledge while adapting to new data.

The main contributions of this work are threefold: (i) the development of a pipeline for deploying a flood prediction system that can begin operating from the very first day of data collection—without requiring a pre-existing dataset for training; (ii) the implementation and evaluation of an incremental learning paradigm tailored for dynamic and evolving environmental conditions; and (iii) the design of a camera-based flood detection system capable of adapting to distributional shifts over time. This approach supports long-term monitoring and enhances the robustness and reliability of flood detection in real-world scenarios.

The remainder of this paper is organized as follows. Section 2 discusses related works on sensor-based and vision-based flood detection systems, as well as recent advances in incremental learning. Section 3 provides an overview of the proposed system architecture. Section 4 details the dataset, training strategy, and evaluation methodology. Section 5 presents the experimental results and analysis. Finally, Section 6 concludes the paper and outlines directions for future work.

2. Related Works

Traditional image-based water level estimation techniques typically rely on fixed visual markers and handcrafted features [Dou et al. 2022]. More recent approaches leverage deep learning for semantic segmentation, aiming to detect flooded areas with high precision [Negrão et al. 2024]. However, these models often assume static deployment conditions. In contrast, incremental learning methods have emerged to address dynamic, non-stationary environments like those observed in long-term urban monitoring systems [Mulimani and Mesaros 2024].

2.1. Image-Based Water Level Estimation with Physical Ruler References

Dou et al. [Dou et al. 2022] proposed a CNN-based water-level recognition method using visual features of ruler markings in river images. Their pipeline included preprocessing steps such as grayscale conversion, edge detection, and morphological operations. Although effective, the method relies on fixed physical rulers, which may degrade or vary across sites.

Zhang et al. [Zhang et al. 2019] tackled similar challenges using the Maximum Mean Difference (MMD) technique to improve detection under complex lighting. However, their approach also depends on standardized water gauges, limiting scalability.

While both methods perform well in controlled environments, they require extensive setup and are unsuitable for dynamic, unstructured settings. In contrast, our approach eliminates the need for physical references, offering a more scalable and adaptable solution for real-world flood monitoring.

2.2. Urban Flood Detection with Deep Learning and Image Segmentation

Recent works by Zeng et al. [Zeng et al. 2024] and Negrão et al. [Negrão et al. 2024] apply semantic segmentation for flood detection using models like DeepLabv3+, Mask R-CNN, and SAM. While Zeng et al. focus on high-precision segmentation, Negrão et al. propose a zero-shot approach for water level estimation without training data.

Although effective, both methods assume static conditions. Our work advances this line by incorporating incremental learning to adapt to environmental changes over time, enhancing robustness in real-world deployments.

2.3. Incremental Learning in Dynamic Environments

Incremental or Lifelong Learning (IL) is crucial in dynamic scenarios where data distribution changes over time, as in our case with evolving vegetation, infrastructure, and camera viewpoints.

Following the Domain-Incremental Learning (Domain-IL) setting [van de Ven et al. 2022], our model handles shifts in input distribution while keeping the output classes fixed and without access to domain labels during inference.

Although Domain-IL has been applied in fields like object detection and acoustic scene classification [Mulimani and Mesaros 2024], it remains overlooked in flood monitoring [Tao et al. 2024]. Our work addresses this gap by introducing an IL-based approach tailored for long-term environmental adaptation without full retraining.

We build on principles outlined by Wang et al. [Wang et al. 2024], emphasizing the trade-off between plasticity and stability to ensure robust flood detection in the face of distributional shifts.

3. System and pipeline overview

This architecture was intentionally designed to be simple and efficient, comprising only two main components: an edge device and a cloud infrastructure. By using a fixed-position camera as the sole sensor, the system remains easy to install, low-cost, and low-maintenance, eliminating the need for complex calibration. These characteristics make the solution not only technically effective but also economically viable and robust for real-world deployment in urban environments.

Figure 1 presents an overview of the proposed flood monitoring architecture, which includes both the edge and cloud components. Below, we describe the two core processes: the flood detection pipeline, where the model performs inference, and the incremental learning pipeline, where model retraining takes place. In the diagram, following the elements in alphabetical order illustrates the flood detection flow, while the numbered steps represent the stages of the incremental learning pipeline.

- **Flood Detection Pipeline (FD):** The detection pipeline operates in real-time. As the fixed camera captures an image, the on-device model immediately classifies the current water level. The result is then sent to the cloud server or used to trigger local alarms if necessary. This lightweight and fast process enables immediate responses to changing flood conditions.
- **Incremental Learning Pipeline (IL):** The training and update pipeline follows a continuous cycle that ensures the model remains up to date with environmental domain changes. It begins with the **data collection** phase, in which the edge device captures images and transmits them to the cloud. At the end of the day, a domain expert performs **manual annotation**, labeling all the collected images with the corresponding water level classes as shown in Figure 4. With this labeled dataset, the system proceeds to the **model retraining** phase. The cloud component performs supervised training with the new labeled data combined with the replay buffer to update the model without requiring full retraining from scratch. Once training is complete, the **updated model is deployed** back to the edge device over the network, replacing the previous version. This process forms a continuous

loop: as the updated model is deployed, it resumes data collection on the edge, restarting the cycle. This ensures that the system adapts progressively to changes in lighting, vegetation, and environmental conditions.

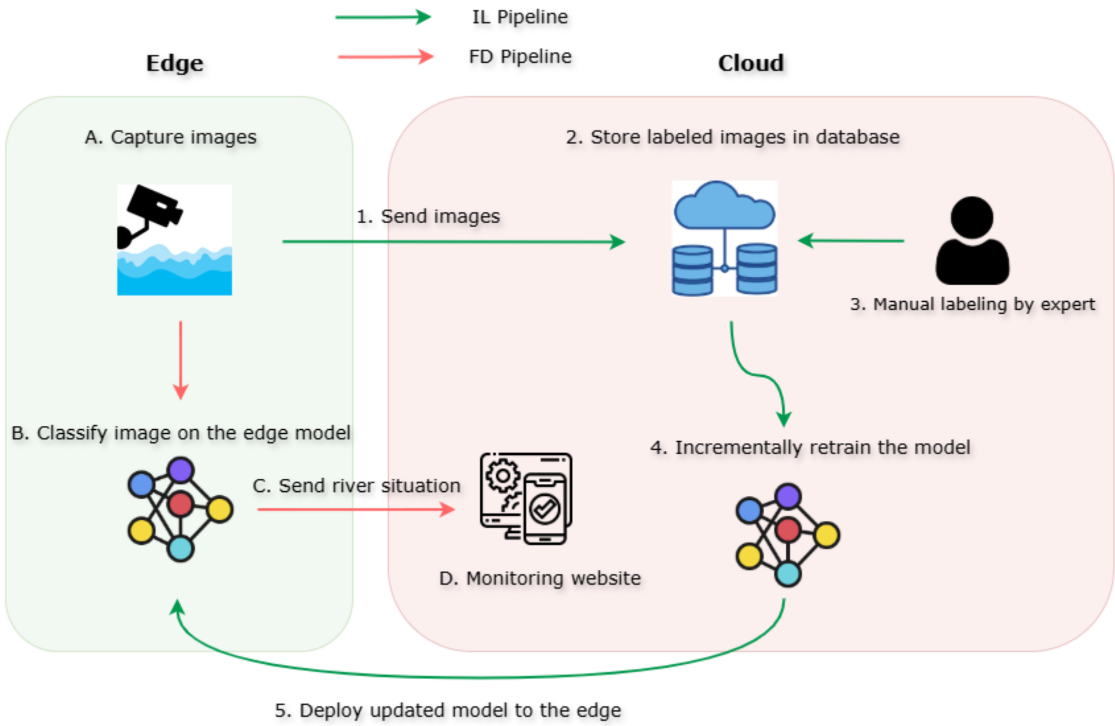


Figure 1. Overview of the proposed flood monitoring system architecture. Steps 1–6 represent the operational flow from data acquisition to model update.

Figure 2 provides a detailed view of how training with the replay buffer works. The new labeled experiences from step $i-1$, together with the replay buffer from $i-1$, form the training set for $i-1$. This set is used to train model $i-1$. After this stage, the replay buffer for i is updated by applying the defined replacement policy to the training set of $i-1$. The process then repeats, with newly labeled experiences forming the training set for i and updating model i.

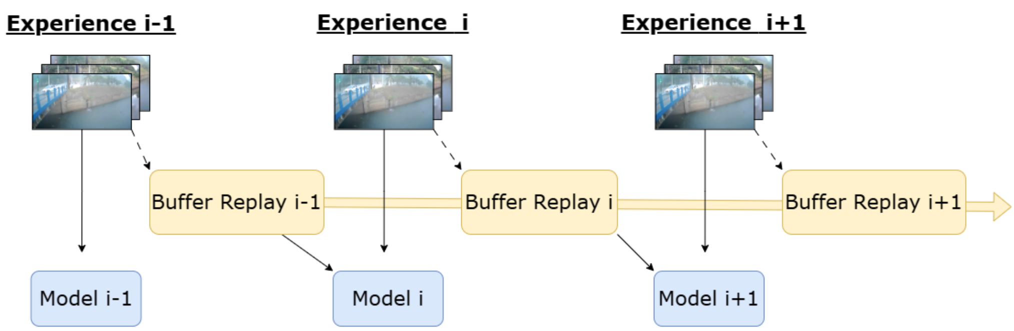


Figure 2. Overview of the training routine used in our incremental learning strategy.

Due to the nature of this pipeline, there are several components that can be explored through ablation studies:

- Different buffer replacement policies and buffer sizes.
- Varying the interval (in hours, days, weeks, or months) between model updates.
- Changing the number of training epochs at each retraining step.

However, in this manuscript, our primary focus is on exploring different buffer replacement policies. Subsection 4.5 describes the specific parameters used in this work for the other aspects, which are not explored here but are left for future work.

4. Materials and Methods

This section presents the materials and methodology used to develop and evaluate the proposed flood detection system. We first describe the dataset collected from an urban stream in São Carlos, Brazil, highlighting its visual diversity and environmental variability.

4.1. Database and problem description

We utilize a single dataset of manually annotated images collected from a fixed monitoring point along an urban stream in São Carlos, Brazil, between October 22, 2024, and January 6, 2025. The images span several months and capture diverse environmental and seasonal conditions. Images were captured at 3-minute intervals, resulting in an average of approximately 420 images per day. However, due to network instability and occasional power outages, the actual number of images collected per day may vary. The dataset comprises 33,089 images, each assigned to one of four ordinal water level classes: low (32,154 samples), medium (625), high (280), and flood (30) (Figure 3). Labels were assigned by a human annotator referencing a level reference lines painted on the concrete wall of the stream that are highlighted in the (Figure 4).

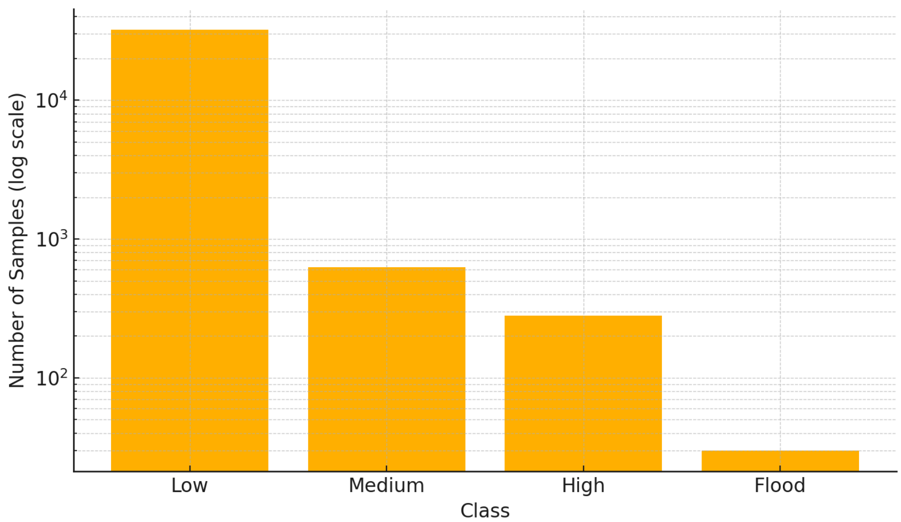


Figure 3. Number of samples in log scale

This dataset exhibits substantial visual variability due to both natural and environmental factors. In this region, rainfall typically occurs in the late afternoon and evening, though it may also happen earlier in the day. Consequently, lighting conditions vary

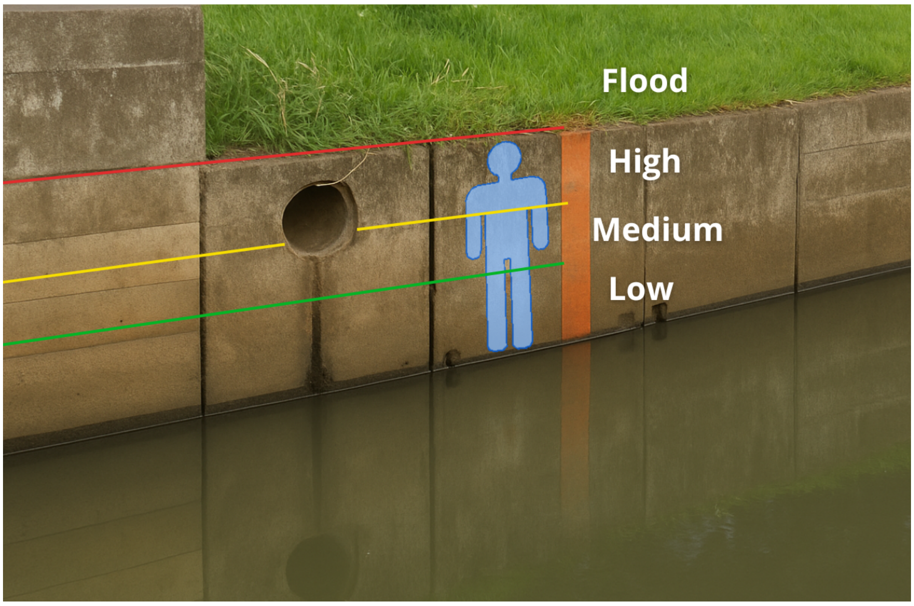


Figure 4. Water level marks on the river walls

considerably across samples. Rainfall intensity and duration introduce further variability within each class. Additionally, the appearance of the water surface is influenced by changes in color, reflections, wind effects, floating debris, vegetation, and nearby objects.

Beyond short-term variability, the dataset captures long-term environmental changes, such as seasonal vegetation growth, infrastructure modifications, and occasional adjustments in camera positioning. These gradual and abrupt shifts pose significant challenges for static models, which may fail to generalize to new deployment scenarios. To ensure the model focuses on the current state of the river landscape, we prioritize training on recent data rather than the entire dataset. To avoid forgetting minority classes, we employ a replay buffer with replacement policies, as described in the following sections. This approach helps maintain robust performance over time.

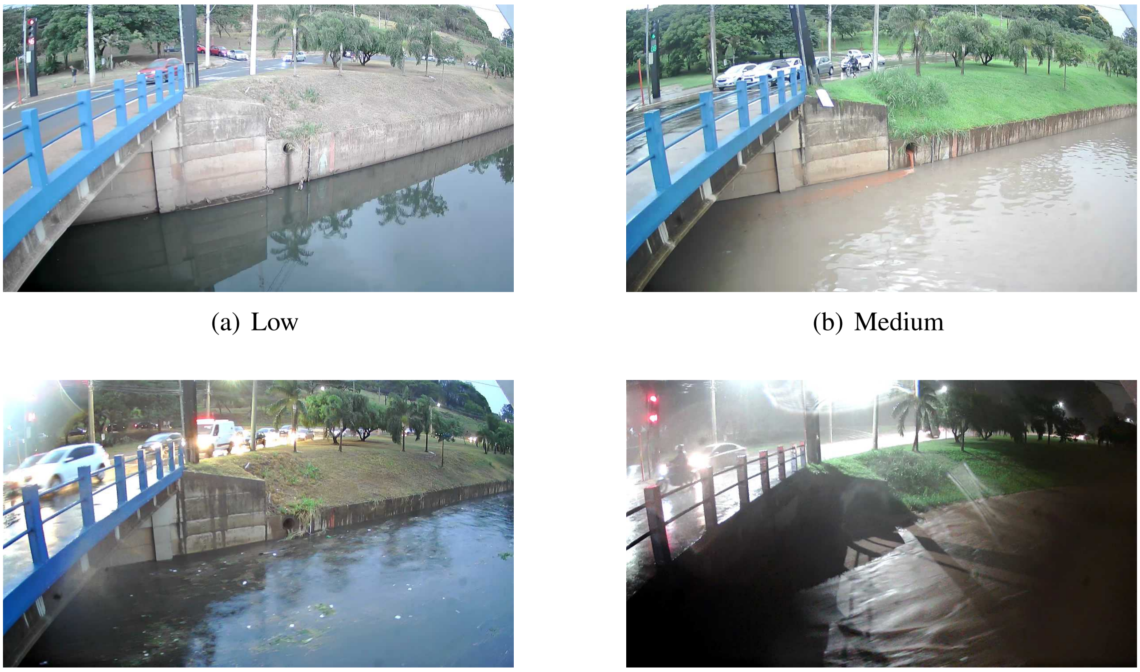
Figure 5 presents representative examples of each class, illustrating the visual diversity captured in the dataset.

4.2. Problem detailing

Using the terminology defined in [van de Ven et al. 2022], our problem is best characterized as a *domain-incremental learning* (Domain-IL) scenario. In this setting, the set of output classes remains fixed throughout training and inference, but the input distribution changes over time due to environmental and visual factors. This requires the model to generalize across shifting domains without access to explicit domain labels during inference.

In our case, the *task* is a 4-class ordinal classification problem based on water level: *low*, *medium*, *high*, and *flood*. Labels are assigned manually using reference markers painted on the wall of a canal.

We define multiple *domains* as distinct visual contexts such as sunny daytime, nighttime, heavy rain, overcast skies, and transitions between weather events. These do-



(a) Low (b) Medium

(c) High (d) Flood

Figure 5. Classes present in the dataset

main shifts naturally occur throughout the dataset and create changes in the visual appearance of the scene.

We refer to the training data received over time as a sequence of experiences E. Each experience corresponds to all images captured during a single day, from 00:00 to 23:59. For instance, one experience may include all images collected on October 23, 2024, and typically spans several domains described above.

4.3. Evaluation Strategy

To contextualize our results, we define two baseline conditions:

- **Lower baseline:** Incremental fine-tuning on new experiences E_i without replay. This represents a worst-case scenario with no mechanism to mitigate forgetting.
- **Upper baseline:** Joint training was performed with access to the entire dataset, simulating a conventional supervised learning setup and serving as an upper bound on achievable performance. The dataset was split into 80% for training and 20% for evaluation. Due to significant class imbalance, we applied targeted data augmentation to the minority classes to encourage balanced learning.
  The augmentation pipeline included random horizontal flips, small rotations, brightness and contrast adjustments, and resizing. Augmented samples were generated for the three minority classes until they collectively comprised 50% of the training dataset, with each class contributing equally to this portion.

4.4. Replay buffer strategies

When data from a new domain arrives and fine-tuning is applied to update a machine learning model, shifts in the input distribution can cause significant weight updates, leading the model to "forget" previously learned knowledge. This phenomenon, known as catastrophic forgetting, is a major challenge in incremental learning. As noted in [van de Ven et al. 2022], the problem is particularly severe in *domain-incremental learning* (Domain-IL), where domain shifts substantially change the data distribution and can drastically degrade model performance.

To address this, we adopt a *replay buffer* strategy. A subset of data from previously seen domains is mixed with new data during training, enabling the model to rehearse past knowledge and maintain performance across domains.

We evaluated two variations of replay buffer replacement policies:

1. **Naive Replay:** This strategy follows a first-in, first-out (FIFO) approach. As new data arrives, it is added to the buffer, and the oldest samples are discarded to maintain the buffer size. This method prioritizes recency, ensuring the buffer is always updated with the latest observations. However, it may discard valuable past examples that are still informative, potentially leading to forgetting of earlier data distributions.
2. **Loss-Based Replay:** In this strategy, the buffer is updated at the end of each epoch by selecting samples with the highest loss, computed using the same focal loss function applied during training. This prioritizes challenging or rare-class examples, though it may lead to an unbalanced buffer if certain classes consistently yield higher losses.

4.5. Model and training configuration

Given the deployment target, an edge device such as a single-board computer or a microcontroller, we selected MobileNet [Howard et al. 2017] as our backbone architecture. MobileNet is a lightweight CNN designed for efficient inference on resource-constrained hardware. It uses depthwise separable convolutions to reduce parameter count and computational load.

All training was performed using the Adam optimizer with a learning rate of 10^{-4}. Baseline models were trained using cross-entropy loss, while incremental learning experiments employed focal loss to better handle class imbalance. Notably, the focal loss was also used to compute the sample losses that guide the replacement policy in the loss-based replay strategies.

Focal Loss [Lin et al. 2017] is particularly effective in handling class imbalance. It modifies the standard cross-entropy loss by introducing a modulating factor $(1 - p_t)^\gamma$, where p_t is the predicted probability for the true class and γ is a focusing parameter. This reduces the contribution of well-classified examples and shifts the focus towards harder, misclassified instances. In our context, it helps the model better learn underrepresented classes, such as *flood*, which are rare but critical for our problem context.

4.6. Evaluation criteria

Due to the strong class imbalance in our dataset, traditional metrics such as accuracy are inadequate, as they tend to favor the majority class. Since our task involves ordinal classification, it is crucial to use metrics that account for class order. As recommended in [Yilmaz and Demirhan 2023], we adopt Cohen's Quadratic Weighted Kappa (QWK), which considers both chance agreement and class ordering through a weight matrix:

$$w_{i,j} = \frac{(i-j)^2}{(k-1)^2}$$

where k is the number of classes.

In addition, we report Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) to measure the magnitude of prediction errors:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad \text{and} \quad \text{RMSE} = \sqrt{\text{MSE}}$$

While MSE and RMSE do not capture label ordering or chance agreement, they complement QWK by quantifying error magnitude.

5. Results

Table 1 summarizes the evaluation metrics for each training strategy, ranked by Quadratic Weighted Kappa (QWK) due to its sensitivity to ordinal class agreement. While MSE and RMSE offered limited model differentiation, QWK provided clearer insights into ordinal performance. The best results for each metric are highlighted in bold. As expected, joint training with full data access yielded the highest overall performance, serving as an upper bound.

Among incremental strategies, the naive replay buffer with 100 samples achieved the lowest error metrics (MSE = 0.061, RMSE = 0.247) but low QWK (0.290), indicating poor ordinal agreement. In contrast, the loss-based replay buffer with 100 samples achieved the highest QWK (0.464) among incremental methods, though with slightly higher error metrics. Buffer size variations had minimal impact, underscoring the importance of the replacement policy.

Normalized confusion matrices (Figures 7 and 6) revealed that joint training produced balanced predictions, while the lower baseline (incremental without replay) suffered severe forgetting. Replay strategies mitigated this effect to varying degrees, with loss-based replay showing better minority class performance as buffer size increased.

Table 1. Comparison of evaluation metrics across different training strategies.

Strategy (Policy, Buffer Size)	QWK	MSE	RMSE
Joint Training (Supervised)	0.927	0.130	0.361
Replay Buffer (Loss-Based, 100)	0.464	0.079	0.282
Replay Buffer (Loss-Based, 200)	0.458	0.077	0.277
Replay Buffer (Loss-Based, 400)	0.455	0.076	0.276
Replay Buffer (Naive, 400)	0.314	0.065	0.255
Replay Buffer (Naive, 200)	0.301	0.065	0.255
Replay Buffer (Naive, 100)	0.290	0.061	0.247
Incremental (No Replay)	0.191	0.096	0.309

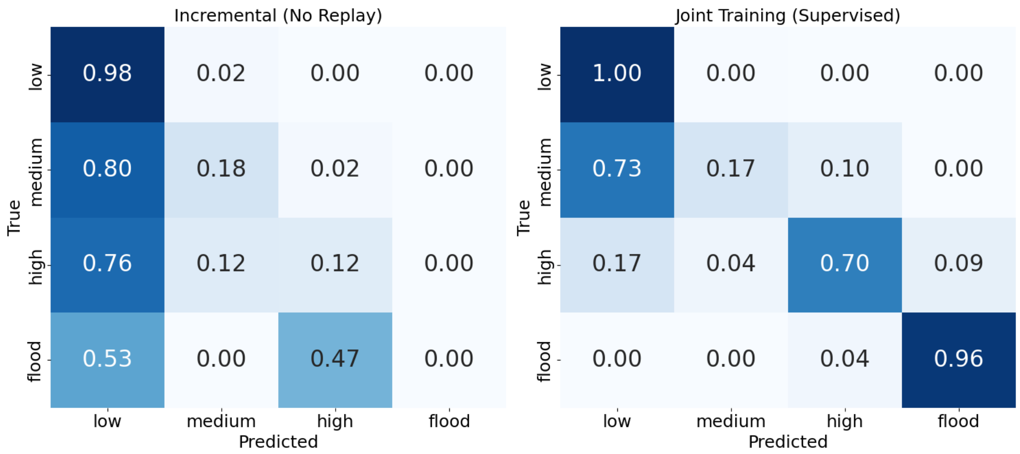


Figure 6. Normalized C.M for Lower (Left) and Upper (Right) baselines.

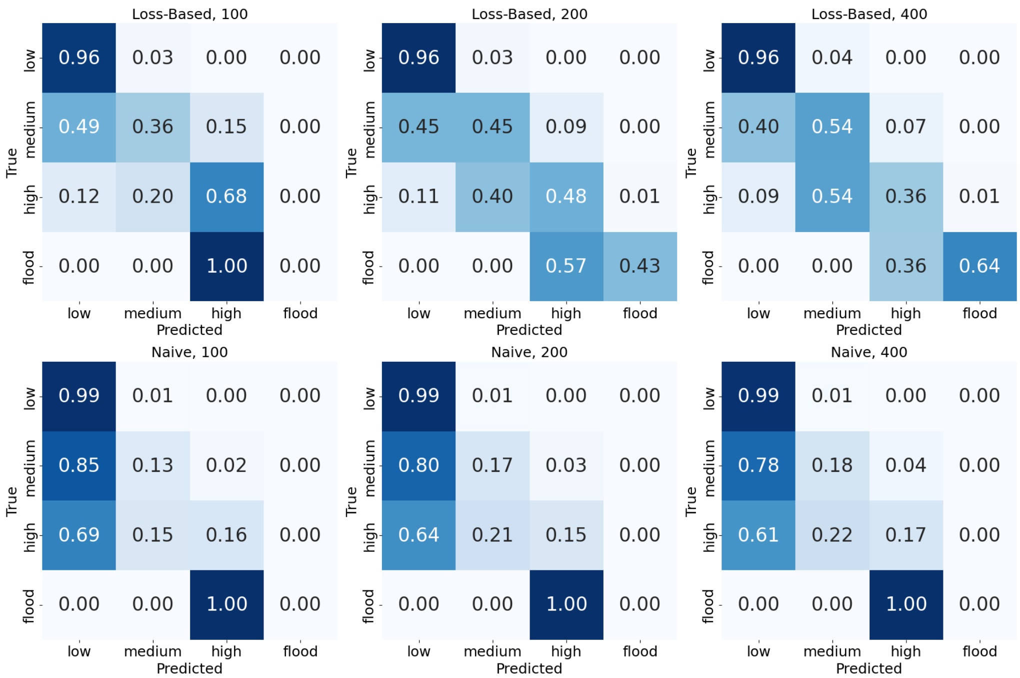


Figure 7. Normalized C.M for Buffer Replay strategies with varying buffer sizes.

6. Conclusions

This work proposes an incremental learning framework for flood detection using image-based water level classification. The incremental learning approach, combined with strategies such as replay buffers, allows the model to continuously adapt to environmental changes, such as shifts in camera position, lighting conditions, and vegetation growth, an aspect that is crucial for this type of system and has not been extensively addressed in recent research.

Experimental results demonstrated that replay buffer techniques effectively mitigate catastrophic forgetting, a common issue in incremental learning systems. By using different replay buffer strategies, such as class balancing, the system was able to enhance its robustness, particularly for minority classes like the flood class, which is critical for public safety.

In future work, collecting more data from different locations and environmental conditions could further improve the evaluation of these strategies. Additionally, exploring architectures that use the most recent image as memory, giving more importance to the immediate previous situation to predict the current one, could help improve classification performance, ensuring greater adaptability and accuracy in flood monitoring systems.

References

Bayram, F., Ahmed, B. S., and Kassler, A. (2022). From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*, 245:108632.

Dou, G., Chen, R., Han, C., Liu, Z., and Liu, J. (2022). Research on water-level recognition method based on image processing and convolutional neural networks. *Water*, 14(12).

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

Machado, G. P. (2024). Floods in south brazil: more than an environmental crisis. *The Lancet*, 404(10447):24–25.

Morelli, A. B. and Cunha, A. L. (2021). Measuring urban road network vulnerability to extreme events: An application for urban floods. *Transportation research part D: transport and environment*, 93:102770.

Mulimani, M. and Mesaros, A. (2024). Online domain-incremental learning approach to classify acoustic scenes in all locations. In *2024 32nd European Signal Processing Conference (EUSIPCO)*, pages 96–100.

Negrão, P. A. S., Ranieri, C. M., Matos, S. N., and Ueyama, J. (2024). Image segmentation with zero-shot generalization for flood prediction in urban environments. In *Proceedings of the 22nd IFAC Symposium on Information Control Problems in Manufacturing (INCOM)*, Vienna, Austria. Forthcoming.

Tao, R., Liang, L., Chen, H., Pan, W., Li, X., Liu, W., and Zhang, H. (2024). A review of cutting-edge sensor technologies for improved flood monitoring and damage assessment. *Sensors*, 24(13):7090.

Tian, S., Li, L., Li, W., Ran, H., Ning, X., and Tiwari, P. (2024). A survey on few-shot class-incremental learning. *Neural Networks*, 169:307–324.

van de Ven, G. M., Tuytelaars, T., and Tolias, A. S. (2022). Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197.

Wang, L., Zhang, X., Su, H., and Zhu, J. (2024). A comprehensive survey of continual learning: Theory, method and application.

Yilmaz, A. E. and Demirhan, H. (2023). Weighted kappa measures for ordinal multi-class classification performance. *Applied Soft Computing*, 134:110020.

Zeng, Y.-F., Chang, M.-J., and Lin, G.-F. (2024). A novel ai-based model for real-time flooding image recognition using super-resolution generative adversarial network. *Journal of Hydrology*, 638:131475.

Zhang, Z., Zhou, Y., Liu, H., Zhang, L., and Wang, H. (2019). Visual measurement of water level under complex illumination conditions. *Sensors*, 19(19).