Applied Intelligence (2024) 54:5536–5557
https://doi.org/10.1007/s10489-024-05466-2

A deep learning workflow enhanced with optical flow fields for flood risk estimation

Caetano Mazzoni Ranieri[1] · Thaís Luiza Donega e Souza[2] · Marislei Nishijima[3] · Bhaskar Krishnamachari[4] · Jó Ueyama[1]

Accepted: 13 April 2024 / Published online: 22 April 2024

Abstract
Owing to the physical and economic impacts of urban flooding, effective flood risk management is of crucial importance. Thus, it is essential to employ reliable techniques for monitoring water levels in urban creeks and detecting abrupt fluctuations in weather patterns. Ground-based cameras alongside a creek offer a cost-effective solution, since they can be deployed for determining water levels through image-based analysis. Previous research has examined the benefits of image processing and artificial intelligence techniques to achieve this goal. However, the current methods only analyze static image features and ignore the valuable motion information that may exist in adjacent frames that are captured minutes apart. In addressing this limitation, our approach involves computing dense optical flow fields from consecutive images taken by a stationary camera and integrating these representations into a deep-learning workflow. We evaluated the capacity of both our method and alternative approaches to measure not only the absolute water level (i.e., whether the water height is low, medium, high, or flooding) but also the relative water level (i.e., whether the water level is rising or falling). The results showed that optical flow-based representations significantly improved the ability to measure the relative water level, while pairs of successive grayscale images effectively determined the absolute water level.

Keywords Computer vision · Flooding · Neural networks · Optical flow · Water level

1 Introduction

Countries worldwide have experienced numerous flash floods in urban areas, sometimes resulting in many fatalities and catastrophic economic losses [1]. Some regions are densely populated with people living in precarious conditions, often on the banks of rivers, which aggravates the deleterious effects of urban floods [2]. However, damage control measures can anticipate and forecast flood events, which are imminent in critical areas, and trigger warning systems to enable the civil defense authorities to handle the event promptly [3].

Flood detection is of the utmost importance for reducing risk in a timely manner [4] and for this reason, numerous flood detection methods have been extensively investigated by academic researchers. These methods include the use of pressure, ultrasonic, and infrared sensors [5–7], which can achieve a high degree of accuracy but require continuous maintenance. Satellite or airborne images have also been explored as potential detection methods [8], although they can prove costly to deploy and are subject to the availability of images, which may not always be readily accessible.

Images from still cameras positioned along riverbanks (i.e., ground cameras) provide a cost-effective and practical solution for flood monitoring and prediction [9]. This system is especially advantageous for assisting vulnerable communities who reside in flood-prone areas. Machine-learning techniques, including neural networks for image segmentation [10] and Convolutional Neural Networks (CNN) [11], have been employed for risk classification or to estimate the absolute level of water bodies on the basis of ground camera images.

In this study, as well as investigating the ability to detect the absolute water level in images, as carried out in previous research studies, we make a significant advance by designing models capable of determining the relative water level (i.e., changes to the water level within a specified time interval). This means of formulating the problem allows us

✉ Caetano Mazzoni Ranieri
cmranieri@usp.br

Extended author information available on the last page of the article

to determine whether the water level has risen, fallen, or remained constant, which might be of particular value in flood risk assessment.

In addition, unlike previous literature, we assess the effects of introducing motion information, extracted from pairs of images taken a few minutes apart, to a deep-learning workflow. We employ dense optical flow fields to improve the representations fed to a convolutional neural network [12, 13], as this approach is similar to the temporal stream employed in video-based human activity recognition research [14–16]. Since the images are captured employing a stationary camera and the sole consistent motion across successive frames is the water surface, this approach is found to be suitable.

Our experiments are based on images obtained from a ground camera and divided into four discrete categories (i.e., low, mid, high, and flooding). The time interval between two frames of images is five minutes. We include an analysis of six alternative input representations integrated into the deep-learning workflow. All these types of inputs are transformations computed from the camera images. These inputs are the following: (i) a single RGB frame; (ii) a pair of successive frames in grayscale; (iii) the absolute difference between a pair of successive RGB frames; (iv) a dense optical flow field; (v) a concatenation between the optical flow and a grayscale frame; and (vi) a stack of temporally ordered optical flow fields.

The remainder of the paper is structured as follows. Section 2 examines the related literature. Section 3 formulates the problem and defines the formulation designed in our analysis. Section 4 analyzes the data and the adopted methodological strategies. Section 5 summarizes the results, and Section 6 discusses their implications. Section 7 brings the study to a conclusion and makes recommendations for future work.

2 Related work

A good deal of research has been carried out on this multidisciplinary worldwide problem, with the aim of finding a means to forecast natural disasters by monitoring the water level in urban creeks and predicting/classifying flood events. The number of studies that address this issue has increased significantly over the past five years [17, 18] and have emerged from different strands and areas of knowledge, such as hydrology, the geosciences, robotics, engineering, and sensors. Hence, many different approaches can be adopted for flood-related problems, and different types of data have been collected and analyzed for this purpose.

Innovations in image processing fields have reduced the cost and time needed to develop and apply these monitoring systems. Moreover, systematic reviews have shown how different image processing methodologies can be applied to flood monitoring and mapping [19], flood risk management [17], and flood forecasting and management [4]. They all include and categorize the applications of Convolutional Neural Networks (CNNs), whether or not they are combined with other methods, for measuring flooded areas, river flow and reservoir levels, as well as predicting river flooding or areas prone to disasters caused by intense rainfall [4, 17, 19]. The authors also classify their data sources as (i) sensors' data, (ii) images from surveillance cameras, (iii) satellite images, (iv) images from unmanned (aerial) vehicles, among others [17].

Customized or hybrid CNNs have been used in flood forecasting as a time series regression model for historical rainfall data [20] with variables such as (i) sensors' measurements as features [21], (ii) a spatio-temporal model in real-time adjacent monitoring stations [22], and (iii) a multiparametric model for spatial input data [23]. More specifically, CNNs have been employed for water level estimation as an image processing method, because they are fed with pipeline outputs such as Hue-Saturation-Value (HSV) color space and morphological processing [24] and mask region features, which are designed for object detection and waterlogging depth measurement [25]. Alternatively, the system can be combined with other methods, such as image feature extraction for other classification methods [10, 26] and pruning algorithms for better generalization [27].

Pan et al. [28] proposed a deep learning-based unmanned surveillance system for observing water levels. The surveillance images enabled the CNN to learn how to extract the Region of Interest (ROI) that was then divided into water and ruler. Following this, it was converted from pixels to the metric system. Vandaele et al. [9, 29] recommended a method based on deep semantic water segmentation to estimate the water level from ground camera images in different UK water bodies. The data was manually annotated with landmarks and flood/non-flood labels through a landmark-based water-level estimation algorithm and used for training large pre-trained neural networks. The DeepLabV3 architecture [30] with a ResNet50 [31] backbone showed the best performance.

Fernandes Junior et al. [32] applied a neural network for semantic segmentation to a flood detection pipeline system based on images obtained from a ground camera that monitored a water creek. The water level was determined through a ground-truth vertical line, and the system would trigger flash flood warnings when a threshold was reached. Qiu et al. [33] applied a dual attention mechanism to obtain global information for semantic segmentation feature maps and corners. An asymmetric convolution with the ResNet50 neural network was used to extract multi-local information that can effectively recognize irregular object sizes caused by angles

stacked with an improved CTransformer. This architecture was designed to retain global contextualized information. It achieved good performance metrics for predicting the level of the water gauge.

The above-mentioned papers are comparable to ours in that they employ methods for determining water levels based on images from ground cameras through CNNs. However, when addressing the problem, none of them took account of the motion information between the images to provide water level measurements. Our approach used subsequent images to quickly estimate the relative measurement required for flood monitoring through dense optical flow [13].

An optical flow field consists of displacement vectors for the motion model between pairs of consecutive frames in a stream. This kind of representation has been useful for extracting features from videos [14, 34]. In his classical paper, Simonyan and Zisserman [35] first proposed the two-stream ConvNet for video activity recognition, which contained a temporal stream based on multiple-frame dense optical flow gradients. Later on, researchers such as Ranieri et al. [16] proposed combining a two-stream ConvNet to data from inertial units and ambient sensors for activity recognition. Ladjailia et al. [36] described a markerless motion-based descriptor using its optical flow features.

Since it is based on a well-known concept to detect motion, optical flow has also been used to determine the flow rate of fluids. McIlvenny et al. [37] compared large-scale particle image velocimetry with dense optical flow when mapping the flow rates in two tidal stream energy sites. Yagi et al. [38] applied dense optical flow to detect river surface flow and compared this system with other techniques. Khalid et al. [39] estimated the motion of river surfaces by means of dense optical flow. Urieva et al. [40] adopted a collision avoidance system for multi-copter UAVs using a simple camera and optical flow as the primary machine vision technique for water obstacle detection.

Compared to existing approaches, the primary novelty of our research is that it sets out a pioneering project for the application of dense optical flow fields for measuring absolute and relative water levels, as formulated in Section 3. By leveraging a deep-learning ResNet50 neural network pipeline, we were able to adopt the original strategy of conducting a comparative analysis of the modalities of input devices rather than solely of architectural nuances. Hence, we were able to make a detailed comparison with the different transformations made to the data before being applied to the deep learning model, while at the same time underlining the effects of dense optical flow representations for the task.

We also made a research contribution by designing two problem formulations (see Section 3) and setting a benchmark with an annotated dataset, which we have made available along with the code for reproducing our experiments. Hence, our work is also unique as it defines relative water levels based on temporally ordered variations of discrete measurements of an urban creek. We hope this endeavor will inspire other researchers to explore further image processing and artificial intelligence techniques.

3 Problem formulation

The models that have been designed and evaluated in this research take, as inputs, images from cameras placed beside urban creeks, which periodically register the images from the water surface and its surroundings. The images might be obtained through the system that is illustrated in Fig. 1: (i) a camera, attached at the top of a pole and connected to a local, embedded system, records the images from the creek and sends them to a cloud infrastructure based on a mobile Internet connection; (ii) the images are registered along with a timestamp; and (iii) a remote workstation is used to access the images and deploy solutions based on them.

Consider a timestamp $t \in T$ for each data frame sent from the system. Let X be the dataset of images and x be the function that links those images with the timestamps, or $x : T \rightarrow X$. Now, we formulate two key problems P_1 and P_2, as follows:

- P_1: determining the **absolute water level** in t_i among a set of ordered categories Y_{abs} corresponding to the water level at timestamp t_i;
- P_2: determining the **relative water level** at t_i with respect to the previous timestamps, among a set of discrete categories Y_{rel} corresponding to whether the water level has fallen, remained still, or risen.

Chronological data were employed to train deep learning models for P_1 and P_2, which were only based on labeled images and other representations derived solely from them. The models shown in this work were designed to approximate the functions $y_{abs} : X \rightarrow Y_{abs}$ and $y_{rel} : X \rightarrow Y_{rel}$. Problems P_1 and P_2 were defined by the composite functions $y_{abs} \circ x : T \rightarrow Y_{abs}$ and $y_{rel} \circ x : T \rightarrow Y_{rel}$. The categories for each problem formulation were defined in (1).

$$\begin{aligned} Y_{abs} &= \{\texttt{low},\ \texttt{medium},\ \texttt{high},\ \texttt{flood}\} \\ Y_{rel} &= \{\texttt{down},\ \texttt{still},\ \texttt{up}\} \end{aligned} \tag{1}$$

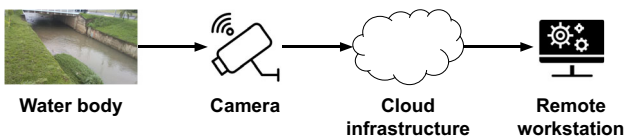


Fig. 1 Illustration of the stages within the proposed global system

The categories in Y_{abs} are ordered so that `low` < `medium` < `high` < `flood`. The categories in Y_{rel} could be obtained by assessing variations in the absolute levels in Y_{abs} across a sequence of timestamps. In the case of two consecutive timestamps t_i and t_{i-1}, the labels $y_{rel}[x(t)]$ were assigned according to the differences between $y_{abs}[x(t_i)]$ and $y_{abs}[x(t_{i-1})]$, as defined in (2).

$$\begin{cases} y_{abs}[x(t_{i-1})] > y_{abs}[x(t_i)] \implies y_{rel}[x(t_i)] \leftarrow \texttt{down} \\ y_{abs}[x(t_{i-1})] = y_{abs}[x(t_i)] \implies y_{rel}[x(t_i)] \leftarrow \texttt{still} \\ y_{abs}[x(t_{i-1})] < y_{abs}[x(t_i)] \implies y_{rel}[x(t_i)] \leftarrow \texttt{up} \end{cases} \tag{2}$$

4 Data and methods

The architectures assessed in this work rely on pictures taken from cameras placed on a pole beside an urban creek. These images are arranged in accordance with the procedures laid out in Section 4.1. The resulting dataset is employed to evaluate different methods for transforming the images before training deep learning models to address $P1$ and $P2$, as shown in Section 4.2. The experimental setup employed to train and evaluate the models is shown in Section 4.3.

4.1 Dataset

In previous work, a data collection system similar to the one shown in Section 3 was implemented in the city of São Carlos as part of the E-Noé project [32, 41, 42]. This system collected images from a critical flooding location at the Mineirinho Creek from 2018 to 2022 and stored them in a cloud server along with their timestamps. The creek is characterized by a very shallow stream whenever the weather is dry. During periods of flash floods, the water level rises fast and can overflow in less than one hour. The region of the creek from which the images were obtained is formed into a canal, and there are walled structures on both of its banks. Each captured image is followed by another one after a time interval of five minutes. This means that every two consecutive images are taken five minutes apart.

Figure 2 shows the average amount of rainfall from 2012 to 2022 in the city of São Carlos, measured at a pluviometrical station located in the Santa Eudóxia district. Data were obtained from the Department of Water and Electricity (DAEE) [1] from São Paulo State. As depicted in the figure, the rainfall events that can cause floods are unequally distributed throughout the year. There are very few rainy days during fall and winter, while the months of November, December, January and February have the heaviest rain. Since our interest

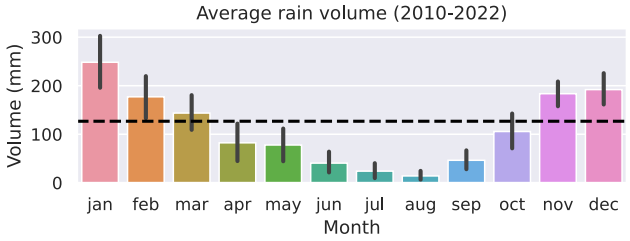


Fig. 2 Average annual rainfall at the system's location. The dashed line corresponds to the mean volume for the whole period

is in variations in the water level that rarely occur outside the rainy season, we decided to use a subset of the images only containing the data from November to February (every day) of the seasons 2018-2019, 2019-2020, 2020-2021, and 2021-2022.

Since our approach relies on supervised learning (see Section 4.2), we had to inspect and label the images manually. Thus, we designed an annotation tool for visually inspecting the 68,599 images for the selected period and assigning a category from Y_{abs} (see Section 3) to each of them.

The guidelines adopted for the annotation procedure are illustrated in Fig. 3. We rely on a marker that was already deployed at the canal and use its references to distinguish low from medium and medium from high. The `flood` label is assigned when the water is above the top of the wall. Random samples of images labeled for each category are shown in Fig. 4.

The annotation procedure yields the class distribution shown in the left panel of Fig. 5, corresponding to P_1 (i.e., absolute water level). The panel to the right shows the class distribution for P_2 (i.e., relative water level), with the labels assigned as expressed in (2).

As expected, the dataset is severely imbalanced in both cases because the water level remains low most of the time and rarely changes. With regard to the absolute level, the majority classes (i.e., `low` and `still`) have a number of images at least two orders of magnitude higher than the other classes. In the case of the absolute water level, the distribution of the `medium`, `high`, and `flood` categories are also imbalanced with respect to each other (e.g., 535 images of medium level and only 75 images of floods). In the case of the relative water level, the `down` and `up` categories are represented by an equitable number of samples.

4.2 The planned approach

Our study involves planning different representations derived from the images through pre-designed transformations and evaluating them as inputs in a machine-learning workflow based on a deep neural network. We prepare our dataset on the basis of Section 4.1 so that it can serve as a use case in which we could assess the proposed models, and particularly

[1] http://www.hidrologia.daee.sp.gov.br/

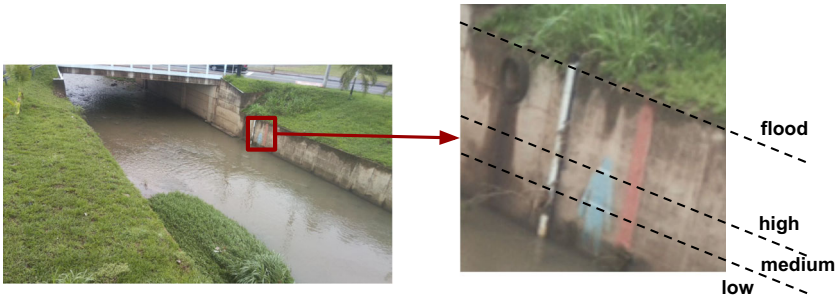


Fig. 3 Annotation guidelines for determining the water level in each image

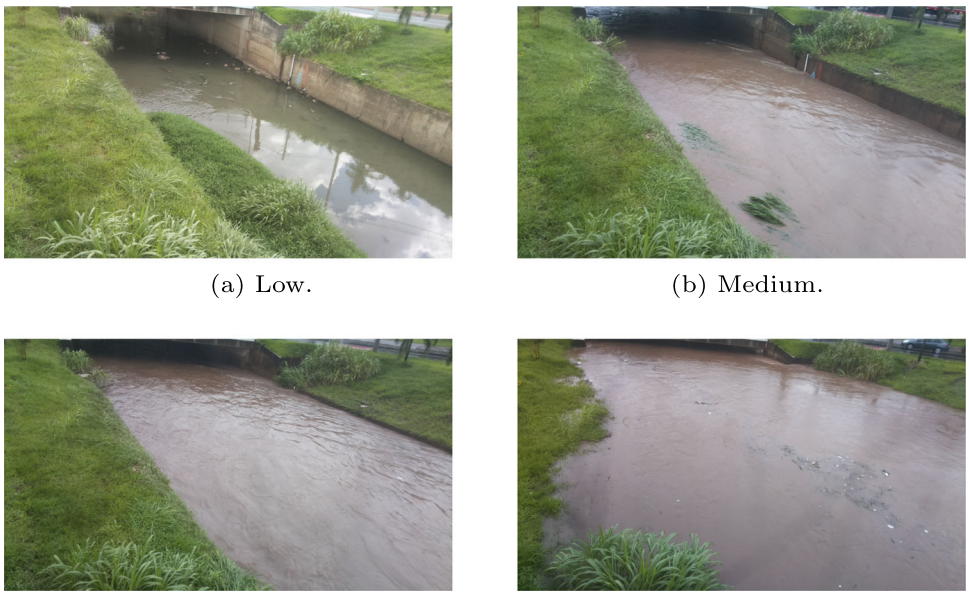
the methods employed for introducing dense optical flow representations [12]. This approach was inspired by works on activity recognition in videos, many of which rely on dense optical flows that are computed across consecutive frames of a video [43]. The following subsections provide details about the methodology of our research.

4.2.1 Dense optical flow

In image processing, motion information in a pair of temporally ordered images can be modeled as displacement vectors at different regions in the first image. This representation of movement as projections of the velocities of its points determines an optical flow field [12]. The computation of the displacement vectors for a set of points of interest results in a sparse optical flow field. Conversely, computing the displacement vectors for all points in the first image is a means of defining the dense optical flow, which we examined in our methodology.

Optical flow computation assumes that a signal region retains approximately the same intensity after a shift in successive frames. This assumption applies to most pairs of frames in a video sequence, in which the objects in the scene and the camera itself usually move only slightly because the frame rate is relatively high (e.g., dozens of frames per second). Several state-of-the-art video classifiers relied on dense optical flow as an additional stream fed to a deep neural network for classification [43].

Although the scenario employed in this study differs because the images are taken five minutes apart, the assumption still holds true because the camera position and the environment surrounding the water body remain almost constant across consecutive images despite the longer time interval. The only feature that is able to consistently move across successive frames is the water surface, whose height is proportional to the variable we seek to measure – i.e., the water level. For this reason, we anticipated that computing the dense optical flow between consecutive images could provide useful information to the machine learning model,



(a) Low. (b) Medium.

(c) High. (d) Flood.

Fig. 4 Examples of images from our dataset, considering each of the categories considered according to our annotation criteria

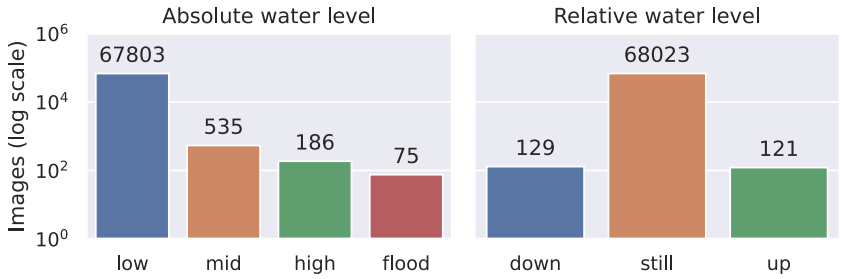


Fig. 5 Number of images for each class after the manual annotation procedure, in logarithmic scale

and thus improve performance, especially for measuring the relative water level.

In our study, we employ the Farnebäck method [44] to compute the dense optical flow for all the pairs of consecutive images registered in the dataset. Two matrices for each image pair represent the resulting optical flow fields, that correspond to the horizontal and vertical components of all the displacement vectors in the images. Once the dense optical flow is computed, there is one displacement vector for each pixel in the first image, and the resulting vector matrices have the same dimension as the input images.

Figure 6 shows an example of this. In this visualization, the optical flow representations are min-max standardized within the $[0 - 255]$ interval and converted to integer values while allowing for a grayscale representation in which the pixel intensity is proportional to the module of the displacement vector.

4.2.2 Input representations

The innovative features of this study result from the different transformations made to the data before it is fed to the deep learning model. We combine operations that directly involve the images, including optical flow-based representations, that form different input representations. Figure 7 depicts the inputs that are only derived from the images through direct operations such as by converting to grayscale or computing RGB differences. Conversely, Figure 8 illustrates the optical flow-based representations.

All the image representations generated, including the optical flow fields, have the same spatial dimensions and only vary in their number of channels (e.g., an RGB image is formed of three channels, that correspond to the intensities of red, green, and blue, while a dense optical flow field is formed of two channels, corresponding to the horizontal and vertical components of the displacement vectors). Since the spatial dimensions are similar for all of the representations included, it is possible to stack different representations of the image as different channels in a single input. The only means adopted in the neural network architecture to account for the different representations is to adjust the number of input channels at the bottom layer.

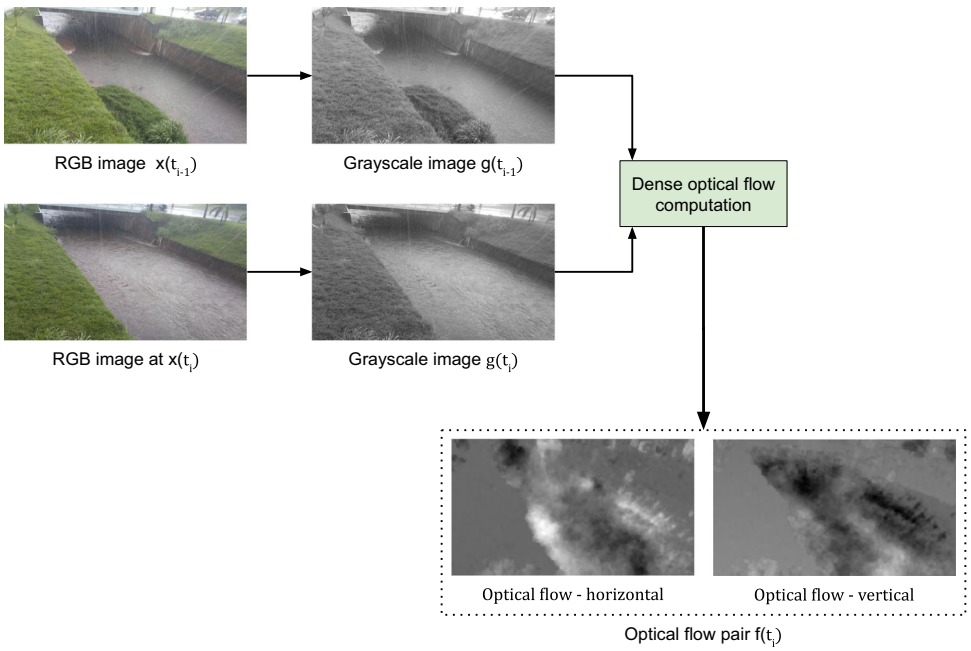
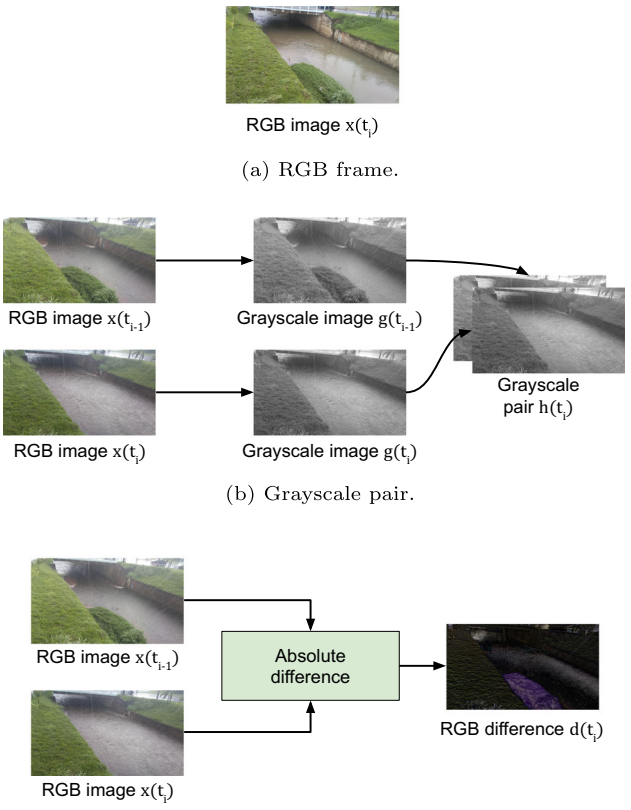


Fig. 6 Example of a pair of consecutive RGB images $x(t_{i-1})$ and $x(t_i)$, their grayscale versions $g(t_{i-1})$ and $g(t_i)$, and the dense optical flow $f(t_i)$ computed from them

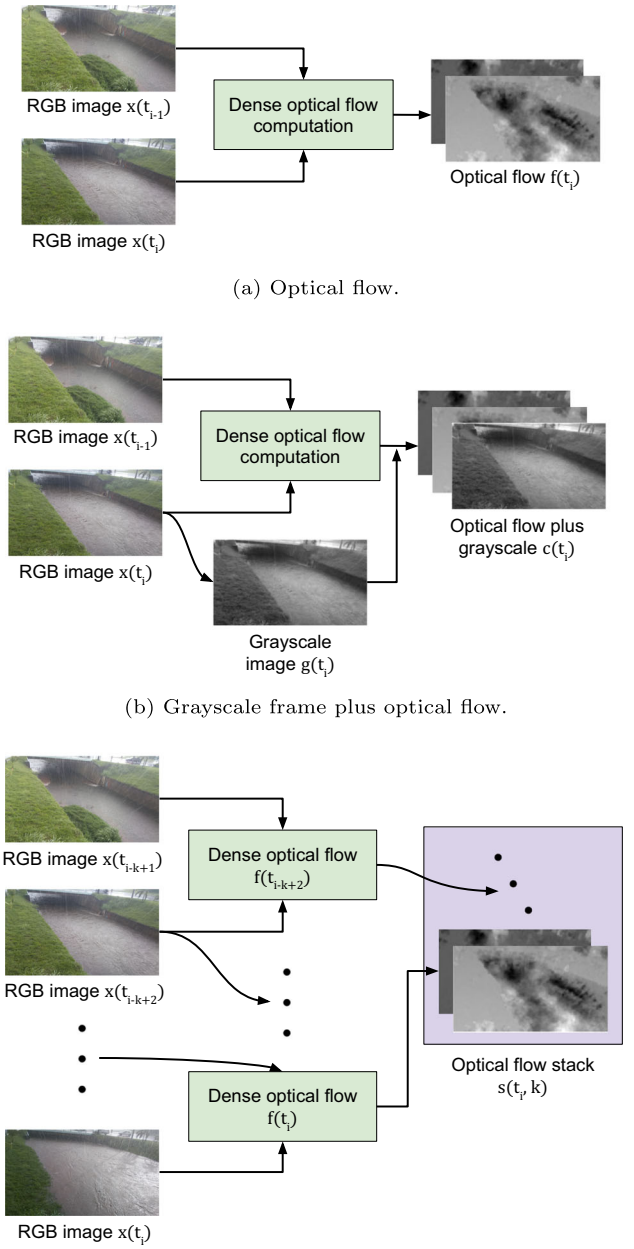


(a) RGB frame.

(b) Grayscale pair.

(c) RGB difference.

Fig. 7 Diagrams illustrating the representations without optical flow at timestamp t_i



(a) Optical flow.

(b) Grayscale frame plus optical flow.

(c) Stacked optical flow.

Fig. 8 Diagrams illustrating the optical flow-based representations at timestamp t_i

Each representation for predicting the absolute or relative water level at timestamp t_i (see Figs. 7 and 8 for the corresponding diagrams) can be described as follows:

- **RGB frame** (Fig. 7a): a one color frame $x(t_i)$ is used as the input representation for the machine learning model. Since this is a traditional RGB image with three channels, it allows a pre-trained model in the ImageNet dataset [45] to be used for transfer learning [46].
- **Grayscale pair** (Fig. 7b): a pair of successive frames $x(t_{i-1})$ and $x(t_i)$ is converted to grayscale, yielding $g(t_{i-1})$ and $g(t_i)$, and stacked to a representation $h(t_i)$. Since $g(t_{i-1})$ and $g(t_i)$ are formed of a single channel representing the pixel intensity, $h(t_i)$ has two channels.
- **RGB difference** (Fig. 7c): the absolute difference between a pair of successive frames $x(t_{i-1})$ and $x(t_i)$ is computed, resulting in $d(t_i)$. The resulting representation is formed of three channels that correspond to the differences in intensity in each channel (i.e., RGB color intensity).
- **Optical flow** (Fig. 8a): a pair of successive frames $x(t_{i-1})$ and $x(t_i)$ is processed by means of an optical flow computation algorithm, resulting in the horizontal and vertical components of the displacement vectors at each pixel. These components are stacked in a representation $f(t_i)$ of two channels.
- **Grayscale frame plus optical flow** (Fig. 8b): the optical flow between the successive frames $x(t_{i-1})$ and $x(t_i)$ is computed, which generates $f(t_i)$. This representation is stacked with the grayscale version of $x(t_i)$, defined as $g(t_i)$. The resulting structure $c(t_i)$ is formed of three channels.
- **Stacked optical flow** (Fig. 8c): the optical flow components from the pairs of the k successive frames in

$[t_{i-k+1}, \ldots t_i]$ are computed, yielding $[f(t_{i-k+1}), \ldots f(t_i)]$ and all of them are stacked into a single data structure $s(t_i, k)$. This structure is formed of $2k$ channels. In our experiments, we employed $k = 3$.

These representations are all fed to a machine learning workflow for training and evaluation so that the model can be used to provide real-time inferences. The next subsection describes this workflow.

4.2.3 Machine learning workflow

In our experiments, we assess the effectiveness of different input representations, including optical flow-based representations, when addressing the problem of $P1$ and $P2$. To ensure valid comparison, we employ a fixed architecture, the ResNet50 [31], which is widely used in applied research [47–50]. Though alternative architectures may have been employed when seeking performance enhancement, we prioritize comparability over individual performance in the classifier network. Moreover, the machine learning workflow discussed in this section can encompass other neural networks for image classification.

ResNets, or Residual Networks, consist of residual blocks encompassing a stack of convolutional layers. The core components of ResNets are the so-called residual connections. As illustrated in Fig. 9, residual connections combine the input of a residual block with its output, which is then fed as input to the following block. This allows much deeper neural networks to be created and thus makes it possible to achieve better results in tasks such as image classification.

The ResNet50 architecture is an example of a ResNet that has 50 weight layers arranged into blocks. The output of the last block feeds is subjected to average pooling, followed by a fully connected output layer with a softmax activation function. Figure 10 provides details about each block and its layers.

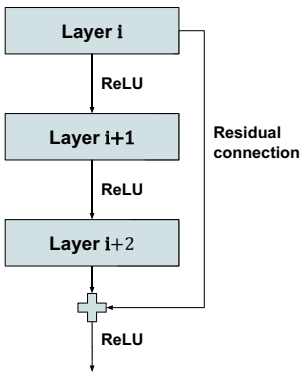


Fig. 9 Residual block with three weight layers

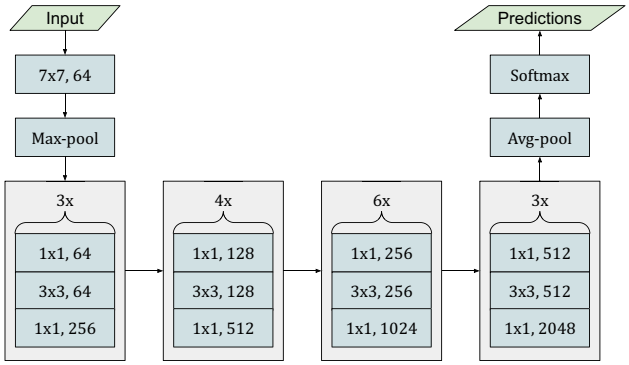


Fig. 10 ResNet50 architecture

In this work, all the inputs described in Section 4.2.2 are taken into account, and the same architectures are trained and evaluated for P_1 and P_2. This allows comparisons between the performance of each model in each problem formulation to be made. The workflows for training and inference are shown in Fig. 11.

In the training phase (Fig. 11a), the extreme class imbalance depicted in Fig. 5 is addressed by implementing a set of standard techniques that are often used for problems of this sort. The same procedures are employed for P_1 and P_2. The majority class is downsampled to 2,000 inputs at the beginning of each epoch. The samples from these classes display low variability, mostly due to the weather conditions and the hour of the day. Hence, the downsampling procedure is liable to little loss of information. In addition, the other classes are upsampled to 2,000 inputs. Although upsampling usually has the same effect as class weighting, we decided to use this technique together with data augmentation, as it can be a source of variability in the training samples.

The batch sampled at each step within an epoch is submitted to a set of data augmentation techniques depending on the type of input. Data from all the representations are processed with random rotations up to 27°clockwise or counterclockwise, translations up to 10% horizontally or vertically, and zooming up to 10% in or out. The *RGB frame* and *RGB difference* inputs are also processed with random contrast adjustments up to 0.1 at each color channel. At each training step, the predictions are used to compute the loss function (i.e., the categorical cross-entropy) fed to an optimization algorithm to adjust the model weights.

The inference phase (Fig. 11b) consists of feeding a preprocessed data sample, in one of the representations presented in Section 4.2.2, to the corresponding model, which is a ResNet50 neural network. The predictions are made in accordance with the input data and the trained model.

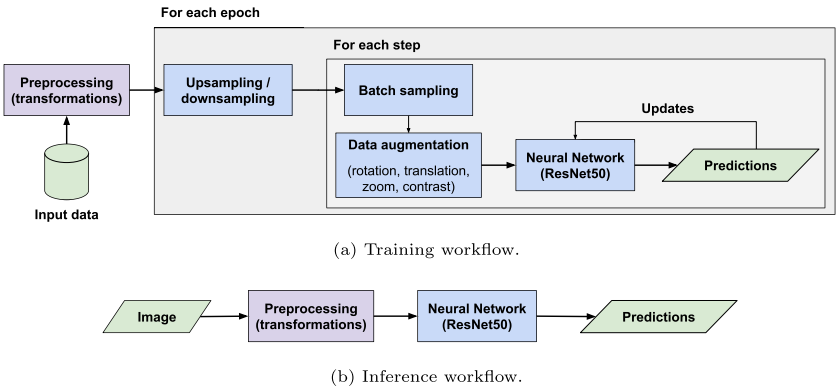


Fig. 11 Machine learning workflows for training and making inferences based on the models

(a) Training workflow.

(b) Inference workflow.

4.3 Experimental setup

Since the dataset consists of four rainy seasons (see Section 4.1), we design four splits for training and evaluating the models, as shown in Table 1. One season is used for testing, and the remaining three, for training. The overall results are obtained by concatenating the predictions of all the test splits. The evaluation metrics are depicted in Section 5.

It can be assumed that the data within each split is independent of the others despite their temporal ordering. This can be explained by looking at the water stream of the creek, which has a steady flow with a low height that only changes during rainfall. When it rains, the water level rises, which can cause flooding, but it returns to its original low level within a couple of hours. Between rainfall events, when the creek is back to its original state, it remains unaffected by past rainfalls. This means that even a time series-based approach can only handle temporal dependencies within each rainfall event.

Despite the short duration of the rainfall cycles, we are still cautious. A splitting procedure is followed that divides the data into blocks of four months that correspond to complete rainy seasons spaced eight months apart. This method ensures the independence of the data points, as no progressive changes related to flooding events can be inferred on the basis of these data points. Moreover, our models rely on single images, pairs, or stacks of images with no more than three images covering a short period.

Table 1 The splits included for the training/evaluation protocol. Each season begins on the 1st of November and ends at the end of February in the next year

	Training seasons	Test season
Split 1	(2019-2020), (2020-2021), (2021-2022)	(2018-2019)
Split 2	(2018-2019), (2020-2021), (2021-2022)	(2019-2020)
Split 3	(2018-2019), (2019-2020), (2021-2022)	(2020-2021)
Split 4	(2018-2019), (2019-2020), (2020-2021)	(2021-2022)

The hyperparameters employed for training the deep learning models are summarized in Table 2. The inputs are resized to 224×224 before being fed to the ResNet50 model. Models pre-trained on the Imagenet dataset [45] are only used for the *RGB frame* modality. In the case of the other types of inputs, the models are trained from scratch. A dropout of 30% is introduced to the top layer.

The loss function employed for training the models is the categorical cross-entropy. The ADAM optimization algorithm is employed for training the models with a learning rate starting at 10^{-2} and declining exponentially after epoch $q > 10$, as defined in (3), where $lr(q)$ is the learning rate at epoch q. The number of epochs is 50, and the batch size is 32.

$$lr(q) = \begin{cases} 10^{-2}, & \text{if } \ q \leq 10 \\ lr(q-1) \times \mathrm{e}^{-0.1}, & \text{otherwise} \end{cases} \tag{3}$$

4.3.1 Performance metrics

Although it is a popular metric for classification models in balanced datasets, the accuracy metric is widely recognized as being unsuitable for evaluating models with imbalanced data [51–56]. Lakshmanan et al. [57] further explain that using accuracy for imbalanced data can lead to misleading

Table 2 Hyperparameters employed for training the deep learning models

Hyperparameter	Value
Input size	224×224
Top dropout	30%
Optimizer	ADAM
Loss function	Categorical cross-entropy
Readout activation	Softmax
Learning rate	see (3)
Batch size	32
Epochs	50

results, as a biased model may achieve high scores without accurately classifying categories other than the majority class.

In our case, over 99% of the samples in the dataset are labeled as low water level. Hence, a dummy model that always predicts the majority class, regardless of the input sample, would result in a degree of accuracy greater than 99%. However, such a model would be unable to identify any of the categories we have prioritized, thus rendering it useless. This means that measuring our results on imbalanced data with the accuracy metric would show that biased models are performing well, while models that can efficiently detect different categories for our problem would be unfairly given lower scores.

Suitable metrics include precision, recall, and the F1-score, which can be computed for each class by (4), (5) and (6). TP is the number of true positive outcomes, TN refers to the true negatives, FP to the false positives, and FN to the false negatives.

$$\text{precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{6}$$

This paper concerns summary metrics computed across all the categories. An intuitive approach for offsetting the class imbalance would entail computing the average value for the per-class accuracy across all the categories. This method is similar to calculating the average of recall, which is also known as the balanced accuracy [52]. Equation (7) defines the balanced accuracy metric, where Y is the set of categories for a particular task and $len(Y)$ is the number of categories.

$$\text{balanced accuracy} = \frac{1}{len(Y)} \sum_{y \in Y} \text{recall}_y \tag{7}$$

Another useful metric is the F1-score, already defined in (6), which consists of the harmonic mean between precision and recall. However, the F1-score is a metric for binary classification. Hence, in this study, we are concerned with the so-called macro-averaging approach, which involves combining the results for all the classes, as defined in (8). This metric involves computing the mean across the individual performance of each class.

$$\text{F1-score (macro)} = \frac{1}{len(Y)} \sum_{y \in Y} \text{F1-score}_y \tag{8}$$

An analysis can be conducted on the basis of the confusion matrix to allow an individual discussion about the performance of a model for each category. The rows in this type of matrix correspond to the ground truth labels, and its columns correspond to the predictions. The arrangement is made so that the value of cell $a_{i,j}$, in which i and j are the indexes of a row and a column, respectively, corresponds to the number of samples from category i that are classified as belonging to category j. Hence, the elements in the main diagonal correspond to the number of correctly classified samples for each class.

In this study, we provide an analysis of the confusion matrices that are standardized by the total number of samples within each class. This gives us the proportion of elements from each category i that are classified as belonging to each category j. In this case, the values in the main diagonal correspond to the separate accuracy for each class.

We also analyze the confidence scores of the ground truth label for each model. These scores are the output of the softmax layer of the neural network at the ground truth label, a value between 0 and 1, which corresponds to the probability inferred by the model that a given sample belongs to the correct (but not necessarily the predicted) category.

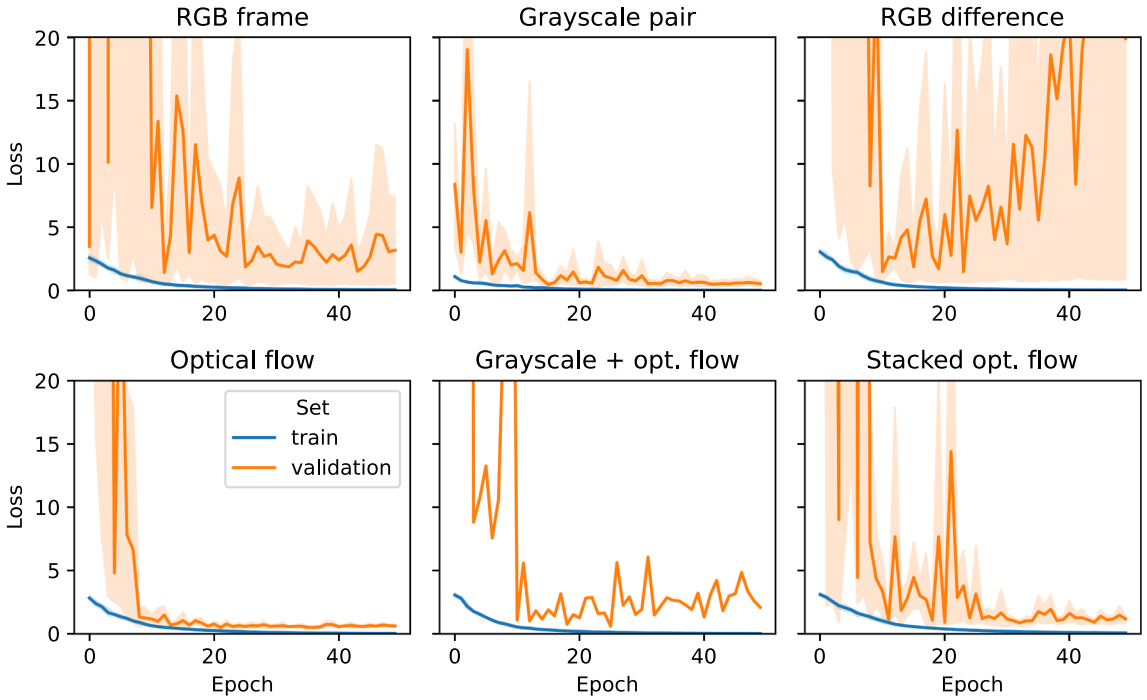
5 Results

To start with, the convergence of the models is shown for each input representation and problem formulation. Figure 12 shows the training and validation loss per epoch, which is averaged across the splits. The shaded areas correspond to the confidence interval.

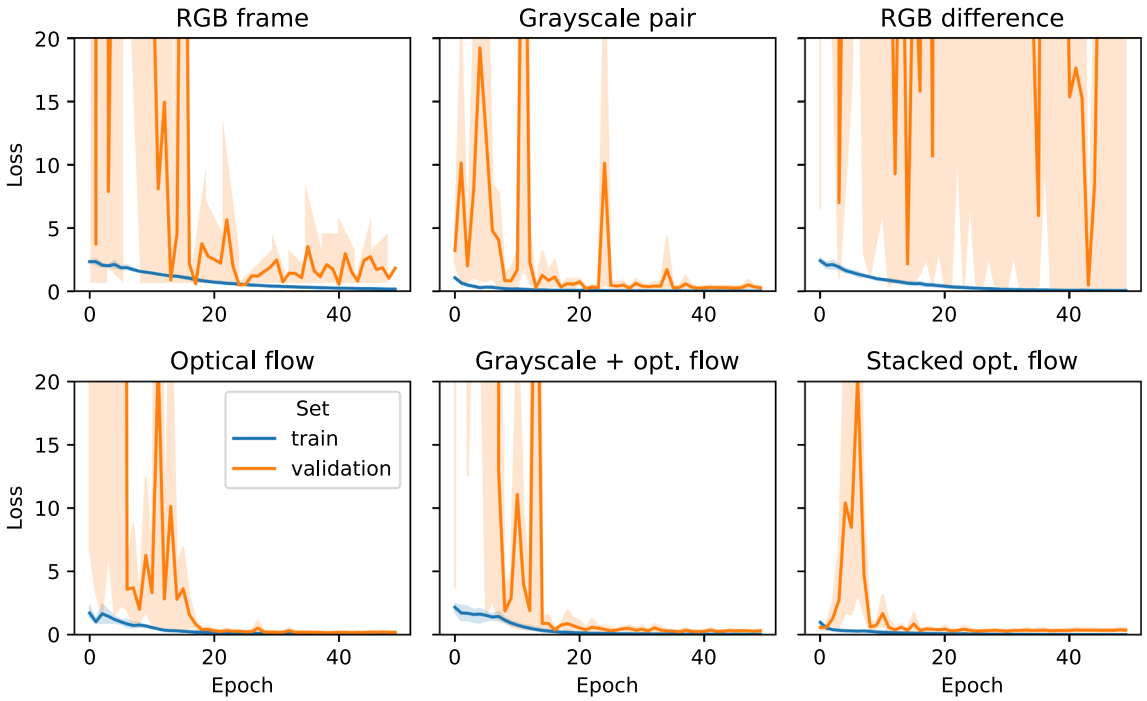
The performance metrics and the corresponding analyses might account for the class imbalance shown in Section 4.1. It should be noted that accuracy, which is a metric commonly employed for balanced datasets, is not very applicable to our case because the majority class corresponds to about 99% of the samples. A bias towards the majority class could benefit models that lack the capacity to classify the minority classes, which correspond exactly to the most important events that the model is expected to predict.

For this reason, as discussed in Section 4.3.1, we adopt the balanced accuracy (i.e., average of recall [58]) as the standard metric summary for evaluating our results. We also take into account the F1 macro (i.e., the average F1-score across all the classes; the F1-score is the harmonic mean between precision and recall). The results for P_1 and P_2 are shown in Tables 3 and 4.

The results for three dummy classifiers are also included as baselines: *majority class*, *stratified random*, and *uniform random*. The majority class classifier predicts the majority class for all the test samples. The stratified random classifier provides a random prediction drawn from a probability distribution that is equivalent to the proportion of samples



(a) Absolute water level (i.e., P_1).



(b) Relative water level (i.e., P_2).

Fig. 12 Train and validation loss for the models for each representation, considering the mean across the splits. Shaded areas correspond to the confidence interval

Table 3 Summary of each of the representative metrics for measuring the absolute water level (P_1)

	Balanced accuracy	F1-score (macro)
RGB frame	64.41%	48.45%
Grayscale pair	**67.88%**	**49.63%**
RGB difference	54.50%	36.94%
Optical flow	53.90%	33.91%
Grayscale + optical flow	63.79%	41.55%
Stacked optical flow	53.73%	35.30%
Majority class	24.73%	24.86%
Stratified random	25.09%	25.08%
Uniform random	23.49%	10.52%

from each class in the dataset. Finally, the uniform random classifier makes a random prediction based on a uniform probability distribution, regardless of the prevalence of each class within the dataset.

Although the balanced accuracy only varied a little for the dummy classifiers, the F1 macro for the uniform random classifier was less than half the value of the others. This result was because, with this classifier, the probability of predicting a minority class is massively superior to its prevalence in the dataset, which results in a large number of false positives for these classes. A large number of false positives affects the precision score and, therefore, the F1-score.

The summary of the metrics provides useful data for assessing the overall model performance measures. Nonetheless, key information for our particular problem can only be assessed by analyzing per-class metrics (see Section 6). Since we are concerned with imminent flooding events, the accuracy in predicting a low or medium water level is not as important as detecting a high water level or a flood. Likewise, determining whether the water level has risen or fallen is more necessary than accurately predicting if it remains still.

Table 4 Summary of each of the representative metrics for measuring the relative water level (i.e., P_2)

	Balanced accuracy	F1-score (macro)
RGB frame	66.96%	34.42%
Grayscale pair	74.42%	47.62%
RBG difference	62.57%	35.44%
Optical flow	80.60%	43.61%
Grayscale + optical flow	71.41%	40.55%
Stacked optical flow	**81.10%**	**56.36%**
Majority class	33.33%	33.27%
Stratified random	33.19%	33.20%
Uniform random	35.10%	16.77%

For this reason, the confusion matrices for the P_1 and P_2 models are included in Figs. 13 and 14, respectively. The values are standardized in terms of the number of samples from each class (i.e., rows add up to one). The per-class accuracy is shown in the main diagonal.

Figure 15 shows the distribution of the prediction scores for a selected set of representations. These scores are the output of the softmax layer of the neural network at the ground truth label t (i.e., the correct class according to the annotations) regardless of its argmax.

In view of the very large number of examples labeled as `low` or `still` (see Section 4.1), which could make the result of the significance test meaningless, we sample m examples from these classes, where m is the average number of examples across the other classes in Y_{abs} and Y_{rel}. This sampling procedure is employed before all the tests are carried out, i.e., the same subsample is used for all the tests.

In the case of P_1 (Fig. 15a), the *RGB frame*, *grayscale pair* and *grayscale plus optical flow* inputs are shown, as they led to the best metrics summary (see Table 3) and also to the highest degree of accuracy for the `flood` class (see Fig. 13).

In the case of P_2 (Fig. 15b), the representations with the best summary metrics are shown (i.e., *grayscale pair*, *optical flow* and *stacked optical flow*, see Table 4). Although the *grayscale plus optical flow* representation led to the best rate of accuracy for the `up` class, we decided to keep one representation without optical flow representations so that its performance could be compared with the optical flow models.

A significance test is also conducted to assess whether the results for each representation differ from each other or if any differences in performance are due to chance. Our experiments are within-subject because the same data were used to test all the models. The subjects are the timestamps of the images, and the factor assessed is the type of input employed in the model. Since the distribution of the scores did not conform to standards of normality, non-parametric tests were employed.

First, the Friedman test is conducted to rule out the null hypothesis, namely, that any differences in the means of the scores for all the representations shown are due to chance. The results of the test revealed a p-value of $p < 10^{-4}$ for all the cases. Afterward, post hoc tests are carried out to check the null hypothesis for each pair of modalities. The Wilcoxon signed-rank test is applied for this purpose, and its p-values are shown in Fig. 15 in star notation[2]. Moreover, the significance level $\alpha = 0.05$ is taken as the standard threshold for significance. In other words, a p-value of $p < 0.05$ leads to a

[2] p-value annotation legend:$(ns \rightarrow p > 0.05);(* \rightarrow p \in [0.01, 0.05]);(** \rightarrow p \in [0.001, 0.01]);$
$(*** \rightarrow p \in [0.0001, 0.001]);(**** \rightarrow p <= 0.0001)$

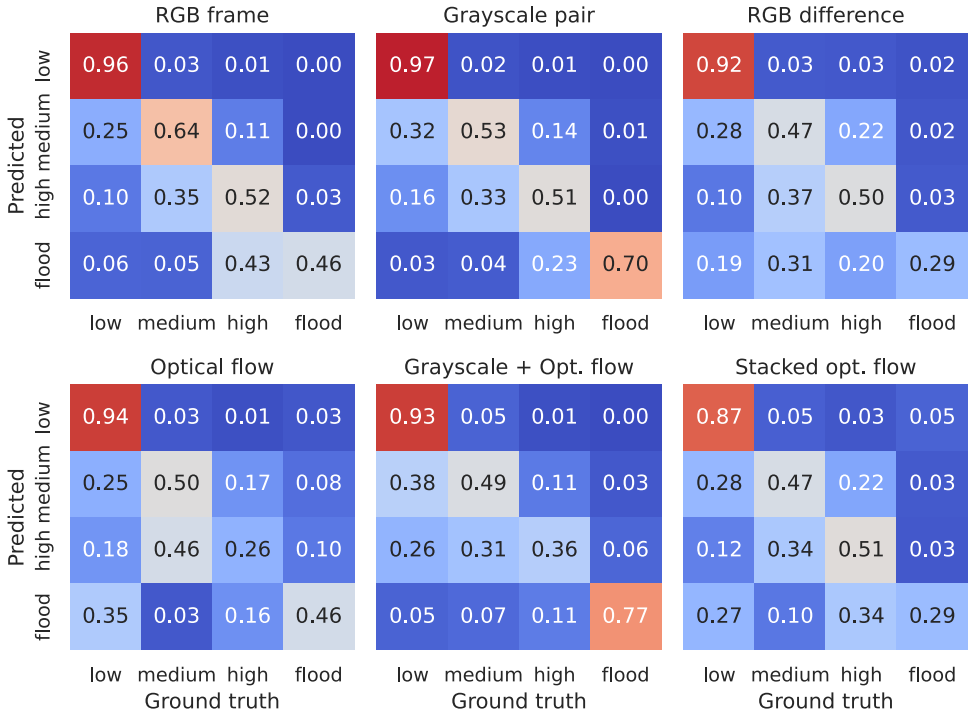


Fig. 13 Standardized confusion matrices of the models for measuring the absolute water level

rejection of the null hypothesis, namely, that any differences in the mean score of the models being compared are due to chance.

We provide classification examples in Figs. 16 and 17. Figure 16 shows the outcomes of the *grayscale pair* model for the absolute water level, as this provides the best performance metrics. We decided to use the *optical flow* model for the relative water level shown in Fig. 17, although the *stacked optical flow* model achieved better results. This was because it is based on pairs of images, which could make it easier for the reader to analyze visually and check how difficult it is to distinguish between discrete levels.

These results show some cases of failure by the models in borderline situations. Although illumination is a key factor,

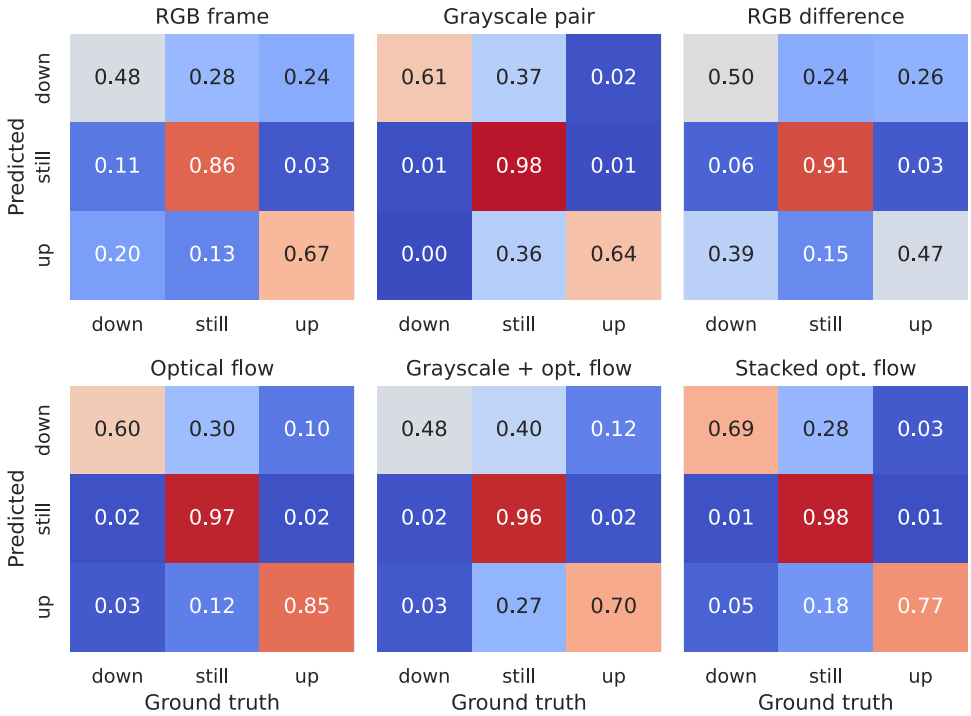
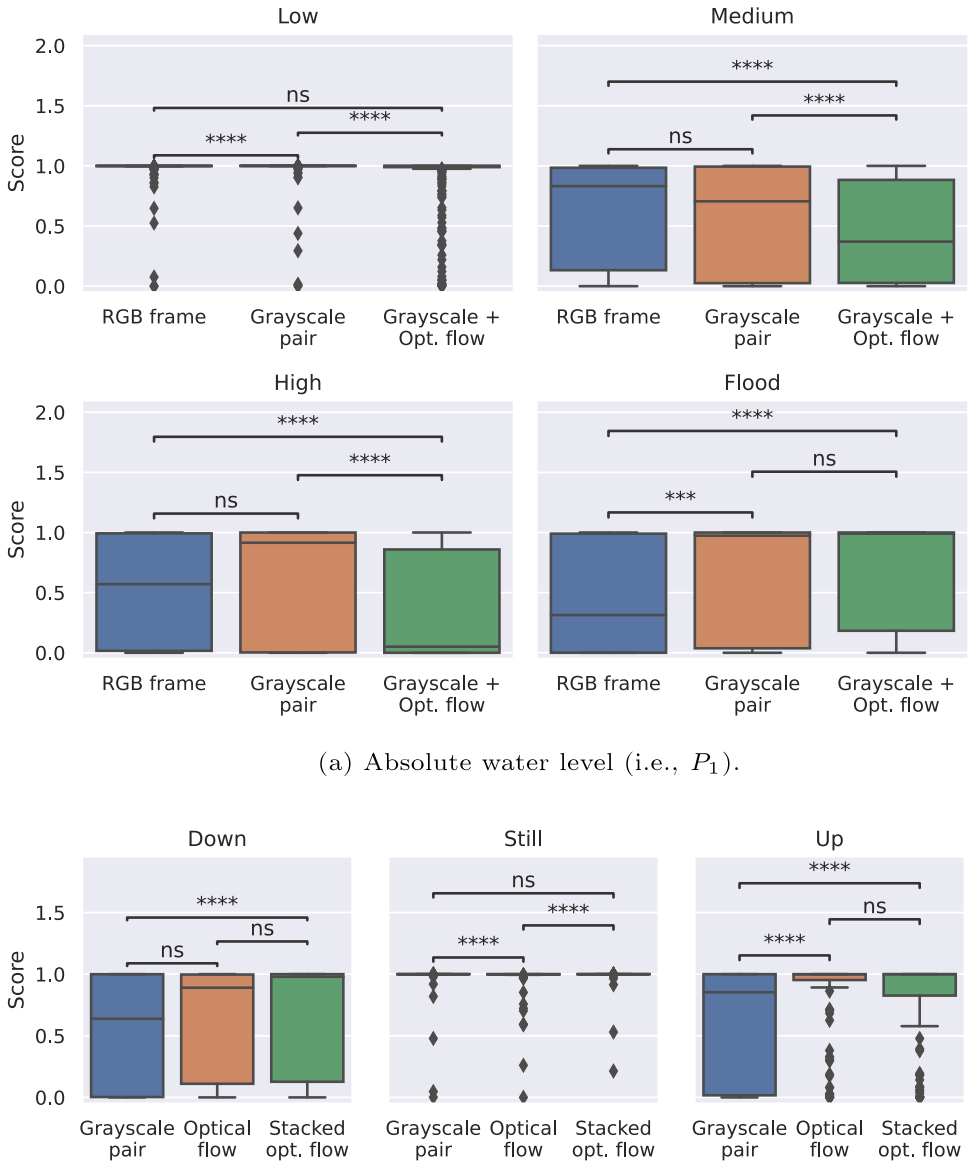


Fig. 14 Standardized confusion matrices of the models for measuring the relative water level between consecutive images



(a) Absolute water level (i.e., P_1).

(b) Relative water level (i.e., P_2).

Fig. 15 Softmax scores of the ground truth class for the representations considered for measuring the absolute or relative water level

many wrong classifications occurred when the water level was very close to the arbitrary level that was used for devising the categories, as shown in Fig. 3.

6 Discussion

As shown in Section 5, the results from the different models trained in this study, each related to one type of input, are assessed with the aid of summary metrics (i.e., balanced accuracy and F1 macro), confusion matrices and the score distributions of the ground truth category.

In the case of P_1, the optical flow fields did not lead to an improved performance when the summary metrics were examined. The results in Table 3 suggest that the grayscale representations responded favorably to the results for this problem formulation since they achieved 67.88% balanced accuracy and 49.63% F1 macro in the *grayscale pair* representation. The *RGB frame* representation also yielded good performance metrics (when compared to the other representations), with 64.41% balanced accuracy and 49.63% F1 macro. The *grayscale plus optical flow* representation, which achieved the best results among the optical flow models, had worse results than the *grayscale pair* (i.e., 63.76 balanced accuracy and 41.55% F1 macro). *Optical flow* and *stacked*

Fig. 16 Classification examples considering the *grayscale pair* model for P_1

optical flow were less accurate for P_1. Their results were comparable to those of the *RGB difference* representation, which yielded poor results for both problem formulations.

One possible cause for these results is that optical flow fields refer to displacement vectors across consecutive images. When inferring absolute water level, the model only applies to the state at a given timestamp. On the other hand, optical flow representations can yield useful information for modeling the relative water level. Indeed, the representations that rely on optical flow fields led to improved performance metrics for P_2, as shown in Table 4.

In the case of P_2, the *stacked optical flow* representation led to the best summary metrics (see Table 4). These were only slightly better than the *optical flow* representation when the balanced accuracy is taken into account (i.e., 81.10% against 80.60%), but there was a more significant improvement for the F1 macro (i.e., 56.36% against 43.61%). The best result among the models without optical flow was achieved by the *grayscale pair* representation, which obtains contextual information from both frames by simply stacking frames $g(t_{i-1})$ and $g(t_i)$ into a representation $h(t_i)$.

In the case of both P_1 and P_2, the *grayscale plus optical flow* representation provided summary metrics that were lower than both the *grayscale pair* and the *optical flow* representations. In other words, the hybrid representation $c(t_i)$, resulting from stacking $g(t_i)$ and $f(t_i)$, led to lower summary metrics when compared with $h(t_i)$, for P_1, or $f(t_i)$, for both P_1 and P_2. However, other outcomes might be obtained by analyzing per-category metrics.

The confusion matrices and the ground truth scores provide a different perspective for analyzing and understanding the results. First, it is worth examining the results regarding P_1, shown in Figs. 13 and 15a. In view of the accuracy of the `flood` category, the *grascale plus optical flow* representation improved the accuracy when compared with *optical flow* by itself (i.e., 77% against 70%), unlike what is shown in the summary metrics (see Fig. 13). This result makes sense if we recall that, in our case study, flood events usually occur after a sharp rise in the water level, which optical flow representations can model. However, the significance tests conducted on the ground truth scores showed no statistically significant difference when compared with *grayscale plus optical flow* and *grayscale pair* representations (see Fig. 15a).

For determining `high` and `medium` water level, the *RGB frame* and *grayscale pair* representations both led to better scores, with statistical significance, when compared with the *grayscale plus optical flow*. When the *RGB frame* and *grayscale pair* representations were compared with each other, there was no statistical significance in determining high or medium water levels, but the *grayscale pair* is more successful in detecting floods. The *grayscale pair* representation was also the best model for determining the `low` water level (i.e., 97% accuracy and statistical significance on the ground truth scores).

There follows a discussion of the results for P_2, shown in Figs. 14 and 15b. The results show that the optical flow-based models led to an improved rate of accuracy for determining the `up` category, with statistical significance. The *optical flow*

Fig. 17 Classification examples considering the *optical flow* model for P_2

representation yielded an 85% rate of accuracy for this category. The ground truth scores of the*optical flow* and *stacked optical flow* representations were higher than the *grayscale pair* representation with statistical significance. However, although the *optical flow* representation led to a higher rate of accuracy than the *stacked optical flow*, this result was not statistically significant.

In an analysis of the confusion matrices, the *optical flow* and *stacked optical flow* models, along with the *grayscale pair* model, showed the best results for the `down` category. With regard to the ground truth scores, no statistical significance was found between *grayscale pair* and *optical flow*, or between *optical flow* and *stacked optical flow*. However, *stacked optical flow* led to a statistically significant improvement for the *grayscale pair*.

With regard to the `still` category, the *grayscale pair* and *stacked optical flow* led to the highest rate of accuracy (i.e., 98%). Both were higher than the *optical flow* representation, which yielded 97% accuracy. The ground truth scores from the *grayscale pair* and *stacked optical flow* were different, with a statistical significance, from *optical flow*. At the same time, no statistical significance was found between them.

Overall, it can be concluded from these analyses that the *grayscale pair* model achieved balanced results for accomplishing P_1, with a high rate of accuracy for all the categories, including `flood`. Although the *garyscale plus optical flow* model achieved the highest rate of accuracy for detecting floods, the differences in ground truth scores for the `flood` category between this model and the *grayscale pair* were not statistically significant. This result suggests that the dense optical flow representations are not very useful for detecting the absolute water level. However, a sequence of grayscale pairs stacked into the representation $h(t_i)$ yielded the best results even with a model trained from scratch (the

RGB frame model being pre-trained on the Imagenet dataset (see Section 4.2).

As regards P_2, the *optical flow* model achieved the best rate of accuracy for the `up` category, with statistical significance for the differences in the ground truth scores when compared with the *grayscale pair*, but not with the *stacked optical flow*. This model also recognized the `down` category with a comparatively high rate of accuracy, and there was no statistical significance in the ground truth scores when compared with *stacked optical flow*. The accuracy for the `still` category was only slightly lower for the *optical flow* model.

An analysis of the validation loss of the models in Fig. 12 shows that the optical flow-based models trained for $P2$ had a smoother and more consistent loss reduction than the *grayscale pair* model. The same was true for $P1$, although the *grayscale pair* resulted in the fastest and most consistent rate of reduction. On the other hand, *grayscale plus optical flow* and *stacked optical flow* converged less smoothly than they do for $P2$. Despite this, these models were more stable than the RGB-based models. It should be noted that the validation loss of the *RGB difference* did not converge.

It is worth pointing out that our supervised models are trained on the basis of weak annotations derived from arbitrary criteria. Marks on the wall of the canalized creek are used to distinguish between low, medium and high water levels (see Section 4.1). However, these categories are not completely different if the graphical features in the image close to the threshold points are compared. It may be difficult even for a human annotator, let alone a machine learning model, to distinguish between low and medium water levels or between medium and high. Thus, the model cannot be expected to be perfectly accurate when the problem is framed as a classification task relying on such weak annotations.

The annotations for the relative water level are inferred from those of the absolute water levels. This inference is based on the assumption that any increase in the water level that crossed one of the references in the wall for two successive images might be labeled as `up`, even if the water level only rises slightly. The contrary may also be true: a slight fall in the water level is recognized as belonging to the `down` category, provided a certain threshold is crossed between these two images. Conversely, some significant rises or falls in water level can be labeled as `still` if no thresholds are crossed between the images.

Some examples in Fig. 16 demonstrate ambiguity since determining whether the absolute water level is above or below one of the threshold lines in Fig. 3 is difficult and subjective. Moreover, the difficulty of the situation can be exacerbated by a number of environmental factors, such as season, time of day, and humidity, which can affect the illumination. As illustrated, many wrong classifications actually result from these borderline cases.

Conversely, Fig. 17 shows three examples for classifying the relative water level. Although there are cases in which the changes to the water level can be evident, there are small, gradual variations in many cases, if not most of them.

Better models could be provided with automatic annotations. For instance, if an additional sensor was included in the creek (e.g., a pressure or ultrasonic sensor), their readings could be used as labels for the images. This strategy would provide numerical variables that are in proportion to the river's actual height rather than the relative position of the water surface to an arbitrary threshold. In this setting, the problem might be framed as a regression task.

It is clear from Section 2 that the works differ from each other in terms of data input and types of problem formulation, which can apply to measuring the absolute water level or the relative level in a time series. This is the first work that uses optical flow to extract motion information to directly determine not only the absolute discretized water level, but also its relative measurements with the aim of providing an early warning system for flooding.

Similar work shows that using ResNet50 is suitable and might produce good results for different related tasks [9, 28, 29, 32, 33]. However, detecting floods is a complex problem, dependent on the formulation of the problem and the source of the data, that is: (i) the type of water body monitored; (ii) the images obtained and their annotations (ground cameras, aerial cameras, satellites); (iii) lighting conditions and angles; and (iv) the fact that it is prone to imbalance.

Some works ensured the water level was measured as a continuous, numeric variable associated with a metric system, and had the error rate of their model expressed in regression metrics such as the Mean Absolute Error (MAE) [28, 32]. Others formulated a binary classification problem (i.e., flood/non-flood) in an unmaintained and unbalanced dataset [9, 29]; Qiu et al. [33] employed a codified system that translates their specific ruler to a readable measurement. Due to these differences in problem formulation and evaluation criteria, the quantitative results obtained from previous studies cannot be directly compared to ours.

As no other models were deployed in similar datasets in the literature, our dataset, which we made public along with the code used in our experiments, acts as a starting point to evaluate different types of models that measure the water level in the flow, particularly relative measurements.

One limitation of our method is that it may not be as precise as other field-deployed sensors like pressure transducers or ultrasonic sensors, which can achieve centimeter-level precision without using machine learning. Another limitation is that our supervised models are trained on weak annotations

that are based on arbitrary criteria, which limits their ability to achieve more accurate results. Additionally, manual annotation is required, which can be tedious and subjective.

7 Conclusion

Flood risk management is becoming increasingly important in different parts of the world, especially in light of the effects of climate change. Damage control measures often rely on the early detection of hazardous events that require urgent attention, such as serious floods. In this study, we are concerned with determining the water level of urban water bodies by only using images. A dataset created in the city of São Carlos, Brazil, was employed in our experiments. This dataset consists of images obtained from a ground camera which periodically registers images at a critical point of the Mineirinho Creek.

Our study lays the foundation for building models that could determine the absolute and relative levels in water bodies. The experiments entail feeding different representations (that were derived from the images) to a deep-learning model and evaluating the results. Our key research contribution is to adopt an approach that is usually employed in video activity recognition, namely, computing a dense optical flow across consecutive images and feeding these representations to a neural network for classification. Video streams can comprise several frames per second, while our dataset is formed of images taken minutes apart from each other. Nonetheless, since the camera is placed at a fixed location, the only element that changes during the sequence of successive images is the surface of the water body, which means that the optical flow field can be helpful for our scenario.

Results show that the optical flow fields prove to be first-rate representations for determining the relative water level (i.e., whether the water level rises or falls in a pair of frames). Increments in the water level can be detected with about an 85% rate of accuracy, compared with 64% achieved by the best model that did not rely on this representation. However, other models are able to classify the absolute water level better. The best models for this task relied on their ability to feed the raw RGB image directly to the network or else stack two consecutive images that are converted to grayscale.

In light of the importance of the relative water level for detecting hazardous situations (e.g., a significant rise in the water level), we believe that the dense optical flow can benefit computer vision by predicting floods from the images obtained by ground cameras.

Nonetheless, the optical flow-based representations that make it easier to determine the relative water levels are helpful in circumstances when a stream fills and empties quickly. Other situations might be investigated further since our results are not directly comparable to those obtained from other models in the literature because both the data and the behavioral patterns of the environment are different.

Our method has two key advantages. First, it operates solely in response to sequences of images captured by a fixed camera, which can be suitably integrated into the surveillance system of a city. Second, the model can be trained by having a single label per image, and thus handle the problem as a standard classification task; this involves leveraging convolutional neural networks that are generally used for image classification. This approach contrasts with others in the literature, where segmentation masks and information from other sources must be included. Our system achieves a reliable performance in detecting climate-related hazards, such as rising water levels or floods. In our view, it can become a valuable additional data source for predicting flood risks in urban scenarios.

The most important limitations of our method are twofold. In the first place, it is not as accurate as field-deployed sensors, such as pressure transducers or ultrasonic sensors which can achieve centimeter precision without the need for machine learning. Despite this drawback, our approach can be useful if deploying these kinds of devices is impossible or slow, such as in remote regions that lack proper infrastructure. Additionally, it can be combined with other methods to improve accuracy or used as a backup system to mitigate the risk of failure.

The second limitation is that our supervised models are trained on weak annotations based on arbitrary criteria, which affects their performance metrics. This makes it a challenging task to distinguish between low, medium, and high water levels, even for human annotators, let alone machine learning models. As a result, this method requires tedious, subjective annotation, and its performance metrics cannot be assessed straightforwardly.

In future work, we intend to design additional innovative devices that can obtain new images synchronized with pressure or ultrasonic sensor data. These can be devised to produce automated labels for the images that allow the models to be trained for a regression task. We also plan to generate synthetic images with the aid of 3D modeling software for better generalization. The synthetic data may represent different scenarios, landscapes, behavior, camera angles, and weather conditions and a diverse dataset of this kind can be used to train models that generalize to different water bodies.

Acknowledgements This research was funded by the São Paulo Research Foundation (FAPESP), grants 2021/10921-2 and 2022/09644-7. It was also carried out using the computational resources of the Centre for Mathematical Sciences Applied to Industry (CeMEAI), which was funded by FAPESP, grant 2013/07375-0. Camera and workstation icons in Fig. 1 by ColourCreatype and Notachai Plukjaisuea on freeicons.io.

Author Contributions All authors contributed to the writing of the draft and the analysis of the results. Ranieri also contributed to the methodology, the data annotation and organization, the design of the modalities, and the machine learning modeling. Donega and Nishijima also contributed to the literature review, the design of the experiments, and the statistical analyses. Krishnamachari also contributed to the interpretation and discussion of the results. Ueyama also contributed to the problem formulation, the data gathering, and the methodology.

Funding This research was funded by the São Paulo Research Foundation (FAPESP), grants 2021/10921-2 and 2022/09644-7. It was also carried out using the computational resources of the Centre for Mathematical Sciences Applied to Industry (CeMEAI), which was funded by FAPESP, grant 2013/07375-0.

Availability of data and materials Data is available for download at https://github.com/cmranieri/flood-detection.

Code availability The code is available at https://github.com/cmranieri/flood-detection.

Declarations

Conflict of interest/Competing interests Not applicable.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication All authors agreed to publish this paper in its present form.

References

1. Antzoulatos G, Kouloglou IO, Bakratsas M, Moumtzidou A, Gialampoukidis I, Karakostas A, Lombardo F, Fiorin R, Norbiato D, Ferri M et al (2022) Flood hazard and risk mapping by applying an explainable machine learning framework using satellite imagery and gis data. Sustainability 14(6):3251. https://doi.org/10.3390/su14063251
2. Oladokun VO, Proverbs D, Adebimpe OA, Adedeji T (2023) Handbook of Flood Risk Management in Developing Countries. Routledge, Milton Park, Abingdon-on-Thames, Oxfordshire, England, UK
3. Sood SK, Sandhu R, Singla K, Chang V (2018) IoT, big data and HPC based smart flood management framework. Sustainable Computing: Informatics and Systems 20:102–117. https://doi.org/10.1016/J.SUSCOM.2017.12.001
4. Kumar V, Azamathulla HM, Sharma KV, Mehta DJ, Maharaj KT (2023) The state of the art in deep learning applications, challenges, and future prospects: A comprehensive review of flood forecasting and management. Sustainability 15(13):10543
5. Faulkner D, Warren S, Spencer P, Sharkey P (2020) Can we still predict the future from the past? Implementing non-stationary flood frequency analysis in the UK. Journal of Flood Risk Management 13(1):12582. https://doi.org/10.1111/JFR3.12582
6. Refice A, Capolongo D, Chini M, D'Addabbo A (2022) Improving flood detection and monitoring through remote sensing. Water 14(3):364. https://doi.org/10.3390/w14030364
7. Ranieri CM, Foletto AV, Garcia RD, Matos SN, Medina MM, Marcolino LS, Ueyama J (2024) Water level identification with laser sensors, inertial units, and machine learning. Eng Appl Artif Intell 127:107235
8. Raj JR, Charless I, Latheef MA, Srinivasulu S (2021) Identifying the Flooded Area Using Deep Learning Model. In: Proceedings of 2021 2nd International Conference on Intelligent Engineering and Management, ICIEM 2021, pp. 582–586. Institute of Electrical and Electronics Engineers Inc., London, United Kingdom. https://doi.org/10.1109/ICIEM51511.2021.9445356
9. Vandaele R, Dance SL, Ojha V (2021) Deep learning for automated river-level monitoring through river-camera images: an approach based on water segmentation and transfer learning. Hydrol Earth Syst Sci 25(8):4435–4453. https://doi.org/10.5194/hess-25-4435-2021
10. Gan JL, Zailah W (2021) Water level classification for flood monitoring system using convolutional neural network. In: Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019, pp. 299–318. Springer, Singapore. https://doi.org/10.1007/978-981-15-5281-6_21
11. Wang Y, Fang Z, Hong H, Peng L (2020) Flood susceptibility mapping using convolutional neural network frameworks. J Hydrol 582:124482. https://doi.org/10.1016/J.JHYDROL.2019.124482
12. Beauchemin SS, Barron JL (1995) The computation of optical flow. ACM computing surveys (CSUR) 27(3):433–466. https://doi.org/10.1145/212094.212141
13. Zach C, Pock T, Bischof H (2007) A duality based approach for realtime tv-l1 optical flow. In: Hamprecht FA, Schnörr C, Jähne B (eds.) Pattern Recognition, pp. 214–223. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-74936-3_22
14. Beddiar DR, Nini B, Sabokrou M, Hadid A (2020) Vision-based human activity recognition: a survey. Multimedia Tools and Applications 79(41–42):30509–30555
15. Pareek P, Thakkar A (2021) A survey on video-based human action recognition:recent updates, datasets, challenges, and applications. Artif Intell Rev 54:2259–2322
16. Ranieri CM, MacLeod S, Dragone M, Vargas PA, Romero RF (2021) Activity Recognition for Ambient Assisted Living with Videos. Inertial Units and Ambient Sensors. Sensors 21(3):768. https://doi.org/10.3390/S21030768
17. Iqbal U, Perez P, Li W, Barthelemy J (2021) How computer vision can facilitate flood management: A systematic review. International Journal of Disaster Risk Reduction 53:102030. https://doi.org/10.1016/J.IJDRR.2020.102030
18. Demir V, Yaseen ZM (2023) Neurocomputing intelligence models for lakes water level forecasting: a comprehensive review. Neural Comput Appl 35(1):303–343. https://doi.org/10.1007/s00521-022-07699-z
19. Arshad B, Ogie R, Barthelemy J, Pradhan B, Verstaevel N, Perez P (2019) Computer vision and iot-based sensors in flood monitoring and mapping: A systematic review. Sensors 19(22):5012. https://doi.org/10.3390/s19225012
20. Khosravi K, Golkarian A, Tiefenbacher JP (2022) Using optimized deep learning to predict daily streamflow: a comparison to common machine learning algorithms. Water Resour Manage 36(2):699–716
21. Dong S, Yu T, Farahmand H, Mostafavi A (2021) A hybrid deep learning model for predictive flood warning and situation awareness using channel network sensors data. Computer-Aided Civil and Infrastructure Engineering 36(4):402–420
22. Zhang Y, Gu Z, Thé JVG, Yang SX, Gharabaghi B (2022) The discharge forecasting of multiple monitoring station for humber river by hybrid lstm models. Water 14(11):1794

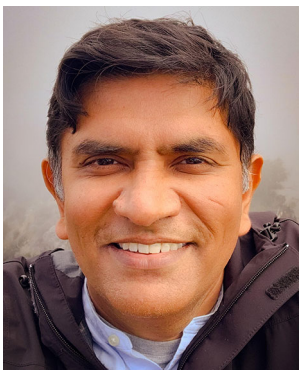
23. Ouma YO, Omai L, et al (2023) Flood susceptibility mapping using image-based 2d-cnn deep learning: Overview and case study application using multipara-metric spatial data in data-scarce urban environments. International Journal of Intelligent Systems 2023
24. Xu Z, Feng J, Zhang Z, Duan C (2018) Water level estimation based on image of staff gauge in smart city. In: 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), pp. 1341–1345. IEEE, Guangzhou, China. https://doi.org/10.1109/SmartWorld.2018.00233
25. Huang J, Kang J, Wang H, Wang Z, Qiu T (2020) A novel approach to measuring urban waterlogging depth from images based on mask region-based convolutional neural network. Sustainability 12(5):2149. https://doi.org/10.3390/su12052149
26. Yang L, Cervone G (2019) Analysis of remote sensing imagery for disaster assessment using deep learning: a case study of flooding event. Soft Comput 23(24):13393–13408
27. Fernandes FE Jr, Yen GG (2021) Pruning Deep Convolutional Neural Networks Architectures with Evolution Strategy. Inf Sci 552:29–47. https://doi.org/10.1016/j.ins.2020.11.009. Accessed 2021-02-13
28. Pan J, Yin Y, Xiong J, Luo W, Gui G, Sari H (2018) Deep learning-based unmanned surveillance systems for observing water levels. IEEE Access 6:73561–73571. https://doi.org/10.1109/ACCESS.2018.2883702
29. Vandaele R, Dance SL, Ojha V (2021) Automated water segmentation and river level detection on camera images using transfer learning. In: Pattern Recognition: 42nd DAGM German Conference, DAGM GCPR 2020, pp. 232–245. Springer, Tübingen, Germany. https://doi.org/10.1007/978-3-030-71278-5_17
30. Yurtkulu SC, Şahin YH, Unal G (2019) Semantic segmentation with extended deeplabv3 architecture. In: 2019 27th Signal Processing and Communications Applications Conference (SIU), pp. 1–4. IEEE
31. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778. https://doi.org/10.48550/arXiv.1512.03385
32. Fernandes FE, Nonato LG, Ueyama J (2022) A river flooding detection system based on deep learning and computer vision. Multimedia Tools and Applications 1–21. https://doi.org/10.1007/s11042-022-12813-3
33. Qiu R, Cai Z, Chang Z, Liu S, Tu G (2023) A two-stage image process for water level recognition via dual-attention cornernet and ctransformer. Vis Comput 39(7):2933–2952
34. Saleem G, Bajwa UI, Raza RH (2022) Toward human activity recognition: a survey. Neural Comput Appl 1–38. https://doi.org/10.1007/s00521-022-07937-4
35. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems 27 (NIPS), Montreal, Canada, pp. 568–576. https://doi.org/10.48550/arXiv.1406.2199
36. Ladjailia A, Bouchrika I, Merouani HF, Harrati N, Mahfouf Z (2020) Human activity recognition via optical flow: decomposing activities into basic actions. Neural Comput Appl 32:16387–16400. https://doi.org/10.1007/s00521-018-3951-x
37. McIlvenny J, Williamson B, Fairley I, Lewis M, Neill S, Masters I, Reeve DE (2022) Comparison of dense optical flow and piv techniques for mapping surface current flow in tidal stream energy sites. Int J Energy Environ Eng 1–13. https://doi.org/10.1007/s40095-022-00519-z
38. Yagi J, Tani K, Fujita I, Nakayama K (2020) Application of optical flow techniques for river surface flow measurements. In: Proceedings of the 22nd IAHR APD Congress, Sapporo, Japan, pp. 14–17
39. Khalid M, Pénard L, Mémin E (2019) Optical flow for image-based river velocity estimation. Flow Meas Instrum 65:110–121. https://doi.org/10.1016/j.flowmeasinst.2018.11.009
40. Urieva N, McDonald J, Uryeva T, Ramos ASR, Bhandari S (2020) Collision detection and avoidance using optical flow for multicopter uavs. In: 2020 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 607–614. IEEE. https://doi.org/10.1109/ICUAS48674.2020.9213957
41. Furquim G, Mello R, Pessin G, Faiçal BS, Mendiondo EM, Ueyama J (2014) An accurate flood forecasting model using wireless sensor networks and chaos theory: A case study with real wsn deployment in brazil. In: Mladenov V, Jayne C, Iliadis L (eds.)Engineering Applications of Neural Networks, pp. 92–102. Springer, Cham. https://doi.org/10.1007/978-3-319-11071-4_9
42. Furquim G, Filho G, Jalali R, Pessin G, Pazzi R, Ueyama J (2018) How to improve fault tolerance in disaster predictions: A case study about flash floods using (iot), ml and real data. Sensors 18(3):907. https://doi.org/10.3390/s18030907
43. Sharma V, Gupta M, Pandey AK, Mishra D, Kumar A (2022) A review of deep learning-based human activity recognition on benchmark video datasets. Appl Artif Intell 36(1):2093705. https://doi.org/10.1080/08839514.2022.2093705
44. Farnebäck G (2003) Two-Frame Motion Estimation Based on Polynomial Expansion. In: Scandinavian Conference on Image Analysis (SCIA), pp. 363–370. Springer, Halmstad, Sweden. https://doi.org/10.1007/3-540-45103-X_50
45. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet large scale visual recognition challenge. Int J Comput Vision 115(3):211–252. https://doi.org/10.1007/s11263-015-0816-y
46. Neyshabur B, Sedghi H, Zhang C (2020) What is being transferred in transfer learning? Advances in neural information processing systems 33:512–523. https://doi.org/10.48550/arXiv.2008.11687
47. Bharati S, Podder P, Mondal M, Prasath V (2021) Co-resnet: Optimized resnet model for covid-19 diagnosis from x-ray images. International Journal of Hybrid Intelligent Systems 17(1–2):71–85. https://doi.org/10.3233/HIS-210008
48. Dawod RG, Dobre C (2022) Resnet interpretation methods applied to the classification of foliar diseases in sunflower. Journal of Agriculture and Food Research 9:100323. https://doi.org/10.1016/j.jafr.2022.100323
49. Wen L, Li X, Gao L (2020) A transfer convolutional neural network for fault diagnosis based on resnet-50. Neural Comput Appl 32:6111–6124. https://doi.org/10.1007/s00521-019-04097-w
50. Oliveira MLL, Bekooij MJ (2022) Resnet applied for a single-snapshot doa estimation. In: 2022 IEEE Radar Conference (RadarConf22), pp. 1–6. IEEE, New York City, NY, USA. https://doi.org/10.1109/RadarConf2248738.2022.9763905
51. Bengio Y, Goodfellow I, Courville A (2017) Deep Learning, vol 1. MIT press Cambridge, MA, USA
52. Brodersen KH, Ong CS, Stephan KE, Buhmann JM (2010) The balanced accuracy and its posterior distribution. In: 2010 20th International Conference on Pattern Recognition, pp. 3121–3124. IEEE
53. Haibo H, Yunqian M (2013) Imbalanced learning: foundations, algorithms, and applications. Wiley-IEEE Press 1(27):12
54. Kubat M, Kubat J (2017) An Introduction to Machine Learning vol. 2. Springer
55. Liu N, Li X, Qi E, Xu M, Li L, Gao B (2020) A novel ensemble learning paradigm for medical diagnosis with imbalanced data. IEEE Access 8:171263–171280

56. Kalid SN, Ng KH, Tong GK, Khor KC (2020) A multiple classifiers system for anomaly detection in credit card data with unbalanced and overlapped classes. IEEE access 8:28210–28221
57. Lakshmanan V, Robinson S, Munn M (2020) Machine Learning Design Patterns. O'Reilly Media
58. García V, Mollineda RA, Sánchez JS (2009) Index of balanced accuracy: A performance measure for skewed class distributions. In: Araujo H, Mendonça AM, Pinho AJ, Torres MI (eds.) Pattern Recognition and Image Analysis, pp. 441–448. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-02172-5_57

Caetano Mazzoni Ranieri is a postdoctoral research fellow at the Institute of Mathematical and Computer Sciences of the University of Sao Paulo (ICMC-USP), with research focused on Artificial Intelligence in the context of the Internet of Things. He graduated in Computer Science at the Sao Paulo State University (UNESP, 2013) and did his Master's degree (2016) and Ph.D. (2021) at ICMC-USP. During his Ph.D., he worked as a visiting scholar at Heriot-Watt University, Scotland (2020). He has experience in Activity Recognition, Deep Learning, the Internet of Things, Machine Learning, and Robotics.

Thaís Luiza Donega e Souza is a PhD in Information Systems at the University of São Paulo. Her research focuses on artificial intelligence for information and experience goods, especially text mining, affective computing, computer vision, and neural networks applied to the cultural and entertainment market.

Marislei Nishijima is a Professor of Economics at the University of Sao Paulo's Institute of International Relations. She holds a Ph.D. in Economics from the University of Sao Paulo and specializes in applied information goods, shocks, and program and policy evaluations. Dr. Nishijima has a strong publication record in academic journals and conferences, reflecting her expertise and contributions to the field.



Bhaskar Krishnamachari is a Professor of ECE and CS at the Viterbi School of Engineering, University of Southern California. He works on emerging technologies like IoT, AI, and Blockchain, and their applications, with an emphasis on networked and distributed systems. He has co-authored three books and more than 300 papers, collectively cited more than 30,000 times. He received his B.E. in Electrical Engineering at The Cooper Union (1998), and his M.S. (1999) and Ph.D. (2002) also in Electrical Engineering from Cornell University. He is an IEEE Fellow.

Jó Ueyama is a Professor at the University of São Paulo, holding a PhD in Computer Science from Lancaster University, United Kingdom. In addition, he is also the Deputy Coordinator of Research for Innovation at the São Paulo State Research Foundation (FAPESP) and a member of the Advisory Committee for Technological Development at the Brazilian National Council (CNPq). He has a publication record with 74 journal articles and over 100 conference papers. He has five filed patents and five software registrations granted by INPI.

Authors and Affiliations

Caetano Mazzoni Ranieri[1] · Thaís Luiza Donega e Souza[2] · Marislei Nishijima[3] · Bhaskar Krishnamachari[4] · Jó Ueyama[1]

Thaís Luiza Donega e Souza
thais.donega@usp.br

Marislei Nishijima
marislei@usp.br

Bhaskar Krishnamachari
bkrishna@usc.edu

Jó Ueyama
joueyama@icmc.usp.br

1 Institute of Mathematical and Computer Sciences, University of São Paulo, Av. Trabalhador Sancarlense 400, São Carlos 13566-590, SP, Brazil

2 School of Arts, Sciences and Humanities, University of São Paulo, Rua Arlindo Bettio 1000, São Paulo 03828-000, SP, Brazil

3 Institute for International Relations, University of São Paulo, Av. Prof. Lucio Martins Rodrigues, s/n, travessas 4 e 5, São Paulo 05508-020, SP, Brazil

4 Viterbi School of Engineering, University of Southern California, Los Angeles CA 90007, CA, USA