

Article

Flow Measurements Derived from Camera Footage Using an Open-Source Ecosystem

Robert Meier ^{1,*}, Franz Tscheikner-Gratl ^{1,*} , David B. Steffelbauer ¹  and Christos Makropoulos ^{1,2}

¹ Department of Civil and Environmental Engineering, Water and Wastewater Systems Engineering Research Group, Norwegian University of Science and Technology, 7491 Trondheim, Norway; david.steffelbauer@ntnu.no (D.B.S.); christos.makropoulos@ntnu.no (C.M.)

² Department of Water Resources and Environmental Engineering, School of Civil Engineering, National Technical University of Athens, Heroon Polytechniou 5, 157 80 Athens, Greece

* Correspondence: robert.meier@ntnu.no (R.M.); franz.tscheikner-gratl@ntnu.no (F.T.-G.)

Abstract: Sensors used for wastewater flow measurements need to be robust and are, consequently, expensive pieces of hardware that must be maintained regularly to function correctly in the hazardous environment of sewers. Remote sensing can remedy these issues, as the lack of direct contact between sensor and sewage reduces the hardware demands and need for maintenance. This paper utilizes off-the-shelf cameras and machine learning algorithms to estimate the discharge in open sewer channels. We use convolutional neural networks to extract the water level and surface velocity from camera images directly, without the need for artificial markers in the sewage stream. Under optimal conditions, our method estimates the water level with an accuracy of $\pm 2.48\%$ and the surface velocity with an accuracy of $\pm 2.08\%$ in a laboratory setting—a performance comparable to other state-of-the-art solutions (e.g., in situ measurements).

Keywords: open source; machine learning; sensors; sewer flow measurements



Citation: Meier, R.; Tscheikner-Gratl, F.; Steffelbauer, D.B.; Makropoulos, C. Flow Measurements Derived from Camera Footage Using an Open-Source Ecosystem. *Water* **2022**, *14*, 424. <https://doi.org/10.3390/w14030424>

Academic Editor: Gwo-Fong Lin

Received: 17 December 2021

Accepted: 25 January 2022

Published: 29 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Traditional sewer flow sensors (e.g., submerged Doppler velocity sensors) are typically expensive and require regular maintenance [1]. Since sewers are hazardous environments, sensors must be robust enough to adequately protect their electronic equipment in the first place if the equipment is meant to operate over long periods. Even then, regular maintenance is needed to keep the sensors operational or at least able to deliver meaningful data. In addition, many countries require explosion proofing—e.g., ATEX [2]—of complex electronic equipment used in sewers, which further increases the costs. These aspects contribute to high initial investments and recurring maintenance expenses, respectively.

Since the direct contact between sensor and sewage, or at least the possibility of it, is the primary source of these costs, the logical solution is to use remote sensing techniques, i.e., sensors which are physically separated from the sewage. Sensors based on ultrasound [3], camera images [4], or infrared [5] can be viable options.

One obvious benefit of using sensors based on image analysis is the possibility for cross-utility of the image material. It is certainly possible to use existing cameras, initially intended for a different purpose, as flow sensors [6], but visual inspection of the flow is simplified if a new camera is specifically installed as a flow sensor. The video feed from such a camera also enables further analyses, for example, to understand the covering layer of fats, oil, and grease [7], detecting blockages or observing the flow regime in general. The manhole condition could even be inspected within the field of view, but only in this fixed—most likely small—area.

Visually determining the water level in a picture essentially boils down to finding the water edge. Therefore, the problem is a form of segmentation, where each pixel in an image is assigned to one of several predefined categories, which typically correspond to objects.

Finding the water edge can be achieved programmatically to a certain degree [8,9], usually in combination with staff-gages acting as objective reference points [5,10,11]. However, more recent approaches typically rely on artificial neural networks [12–14]. Once the border between water and air is known, the pixel distance in the image can be converted back to a physical distance—under consideration of potential image distortion. This approach has been applied successfully in the past to determine the water level in various river settings [15,16] and sewer pipes [4].

Measuring velocity at the water surface (image-based velocimetry) is also possible. The most usual approach to measuring the surface velocity is called particle image velocimetry (PIV), where particles in a stream are tracked using image analysis algorithms. This method has successfully been applied in diverse areas such as discharge measurements in rivers [17–19], shallow overland runoff [20], or sewer discharge measurements [21]. A problem with this method is the availability of trackable particles, which often makes artificial seeding necessary. Artificial seeding relies on the manual addition of clearly visible particles to the water flow, which enables their tracking by the PIV method. Approaches are being explored to compensate for the lack of such particles [22], but are not widely used in practice yet.

This paper demonstrates how footage from cheap, consumer-grade cameras can be used in a real-time flow sensor, without relying on trackable particles. Images of an open channel flow are fed into a convolutional neural network (CNN), which extracts the surface velocity and the water level, which are then used to approximate the flow in real-time. CNNs have gained popularity in image segmentation, where contents of images are automatically detected and classified [23,24]. The main advantage of these networks for this type of problem stems from their ability to detect shapes/objects independently of their exact position in the image (translational invariance). In our case, where flow measurements are conducted based on images, the method is in principle able to generalize to new locations—where the physical layout is not identical to the one the network was initially trained on—without the need for extensive re-calibration.

As the hardware used in our sensing approach is relatively cheap (around EUR 150), it is an ideal option to introduce redundancy where needed, or to determine when existing sensors require re-calibration. Since at least one additional sensor is needed to provide reference values during the initial calibration, introducing it at an existing measurement site to provide redundancy is optimal and does not require a substantial investment. However, if the sensor is to be used at a new location and a reference sensor is needed for the calibration period, the—presumably—costly reference sensor is freed-up afterwards and can be used to put additional measurement sites into operation. This approach can allow for a more complete sensor network to be built, while keeping costs manageable.

Apart from reducing measurement uncertainties, redundant measurements can also serve as an indication when sensor re-calibration is needed. As soon as diverging measurements are detected, the measurement site needs the attention of network operators, thus focusing maintenance efforts.

The presented approach performed well in a laboratory setting, and in an additional step, the effect of several practical restrictions was also explored by scenario testing. As the algorithm's accuracy directly depends on the amount and quality of the training data, the effect of missing data is presented, and practical recommendations for the calibration data collection are given.

2. Materials and Methods

To predict the water level and surface velocity, data in the form of video footage was collected from an open channel in a laboratory setting. Parts of it were then used as training inputs for several different CNNs, while the rest were used to validate the networks' performance. The CNNs were trained from the input images, where one network was given only one of the tasks and never predicted both the water level and the surface velocity

at the same time. Their performance was evaluated in 4 different scenarios, differing in data availability, and compared to the existing measurements.

We will first describe the laboratory setup, where the data was obtained, the limitations induced by it, the used hardware, and then move on to the data pre-processing, predictor training and end this section with a description of the different data availability scenarios used to simulate a more realistic measurement campaign. A summary of the complete workflow consisting of the data gathering (I), finding and training of the best CNN (II) and evaluating the best network for each scenario (III) can be found in Figure 1 below.

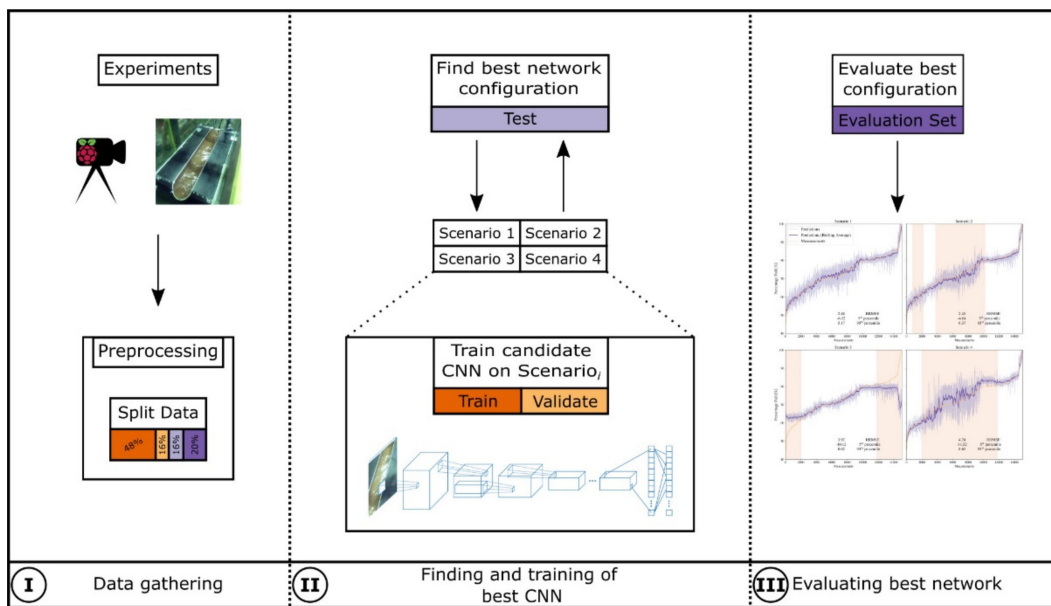


Figure 1. Flowchart of the applied methodology.

2.1. Laboratory Setup

The training data was obtained in a controlled laboratory setting for three different camera positions (see (I) in Figure 1). An open channel consisting of a half-pipe with a radius of 10 cm was placed inside an artificial flume, allowing a convenient way of adjusting the water level inside the area of interest. Two reference sensors, a ‘mic+35/IU/TC’ ultrasonic sensor (accuracy: $\pm 1\%$) and a ‘Nortek Vectrino’ (accuracy: $\pm 1\% \pm 1 \text{ mm/s}$) velocimeter, provided the ground truth values for the water level and the surface velocity, respectively. The video footage was collected at 25 frames per second with a resolution of 720×720 pixels by a Raspberry Pi Camera Module v2 attached to a Raspberry Pi 4 at three different camera positions P_1 , P_2 , and P_3 (see below). Since the footage needs to be converted, stored, and eventually transmitted, we split the video stream into smaller blocks of 15 s. Each block included the timestamp to simplify the subsequent analysis, making the correlation to the ground truth values collected by the other sensors straightforward. We used the following three camera positions:

- P_1 : Directly above the channel ($\theta = 0^\circ$, according to Figure 2)
- P_2 : Above and in front of the channel ($0^\circ < \theta < 90^\circ$ and $\phi = 0^\circ$, according to Figure 2)
- P_3 : Above and to the side of the channel ($0^\circ < \theta < 90^\circ$ and $0^\circ < \phi < 90^\circ$, according to Figure 2).

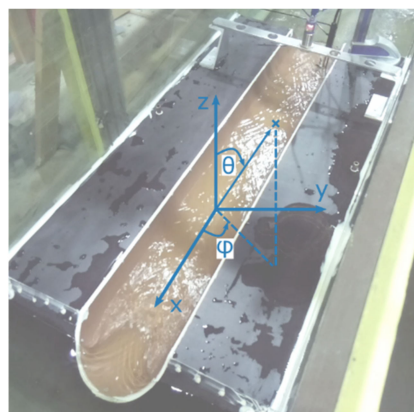


Figure 2. Laboratory setup of the open channel used for the experiments. Different camera positions are described with angles ϕ and θ in relation to the channel geometry. The depicted channel has a length of 1370 mm, and the half pipe has a radius of 100 mm.

2.2. Limitations

Since the reference sensors have fixed positions, the velocity is not always measured directly at the surface, while the water level fluctuates. To gauge this effect, we manually measured the velocity at different depths of a constant flow. The differences are negligible in this laboratory setting.

Similarly, we could not guarantee a perfect sensor alignment to the water surface. Consequently, the flow is not ideally parallel to one of the axes in the sensor reference coordinate system (x_s , y_s , z_s). As described below, we assumed the reference surface velocity to be the ℓ^2 -norm of the measurements along the three sensor axes x_s , y_s and z_s (i.e., the length of the velocity vector).

Due to the geometry of the Vectrino sensor, which measured the surface velocity, it was impossible to measure the surface velocity of shallow water levels. As the water level and the surface velocity correlate, the training set does not contain very low-velocity samples, even though the prediction method should, in theory, not behave differently for low levels/slow flows, although we could not verify this.

We tested the method under visible light and did not use the infrared (IR) spectrum. Using a traditional lamp as a visible light source may be problematic in a field setting due to the attraction of insects/animals. Since the prediction accuracy of the networks does not depend on colour information (tests using different image colouring showed no significant change in the prediction accuracy), using IR light sources could be a good alternative in a practical sewer setting. The primary goal of this project is to target enclosed channels which exhibit constant lighting conditions. However, channels outside are exposed to various changes in lighting, both predictable (e.g., daytime, seasonal) and unpredictable (e.g., weather). This effectively introduces an additional dimension in the training set, which needs to be evaluated in a further analysis.

2.3. Hardware

To keep the cost of the sensor as low as possible, Raspberry PIs were used to collect the video footage. These single-board computers can easily be equipped with additional sensors, such as the cameras used here, which can then conveniently be controlled, for example, through a Python interface. As the platform can run Linux operating systems, it is easy to make exclusive use of open-source software, reducing the cost even further and allowing custom modifications. The use of open-source software not only reduces up-front costs but also prevents vendor lock-ins, which can lead to a non-negligible source of costs in the long run. Currently, the cost of the entire sensor is around EUR 150, making it more affordable than established products. Low costs are vital to making wide-spread sensor deployment possible to move towards more thoroughly monitored sewer networks.

2.4. Pre-Processing

The pre-processing, shown in part (I) in Figure 1, of the training and validation data had the following two main goals: introducing more structure to the data and the elimination of outliers. Adding structure in this context means to correlate frames to the corresponding reference measurements. For that the footage is split into frames and the measurements resampled, as described below.

Using the interquartile range (IQR), extreme outliers are identified by being outside of the so-called lower and upper ‘outer fences’, defined by: $\pm 3 \text{ IQR}$ [25]. These values were removed and interpolated linearly, with the largest gap consisting of 12 consecutive measurements (roughly half a second of footage). All measurements were then resampled using the mean, to a frequency of 1 Hz, as this was the lowest common measurement frequency.

Since the velocimeter measures the movement in three dimensions (x_s , y_s , z_s), the length of the resulting vector was used as the reference surface velocity measurement. Ideally, if the sensor were perfectly aligned with the water surface, calculating the resulting vector would be unnecessary, but since there are always small inaccuracies in the setup, we made this decision. As expected, one axis (in direction of the flow) contributed almost exclusively to the result.

The water level predictor was trained on individual frames, while the surface velocity prediction used 25 consecutive frames as input, i.e., one full second of continuous footage which corresponds directly to the 1 Hz measurement frequency. The images were cropped to focus solely on ones that are of interest, where the channel border and the flowing water is visible. This was done exclusively for the purpose of reducing the input size which in turn reduces the network complexity.

2.5. Predictor Training

A CNN is a special kind of neural network, whose basic idea—when dealing with image data—is to slide, or convolve, filters over its input image. One filter recognizes one shape, for example a horizontal edge, and determines how well each part of the image corresponds to its shape. One stage of the CNN consists of multiple such filters, all looking for potentially different patterns. The result of such a stage is then fed into the next stage, thus applying filters in succession.

As the CNN gets deeper and the inner filters operate on the output of the previous filters, the patterns they detect can be more complex and may even start representing concepts, such as “a tree” compared to simple patterns like “a horizontal line”.

Typically, a CNN uses a small, fully connected neural network after the initial filter stages to interpret their results and map them to the desired output.

To find the best predictor for our problem we evaluated different network topologies which varied in the number of layers and their sizes (see part (II) in Figure 1). All our networks generally consisted of a first part which contained the convolutional layer logic (the filters, as described above). Each convolutional layer was alternated with a max-pooling layer, to continuously reduce complexity. To restrict the number of possible configurations, we considered only networks with 1–5 layers. This first stage was then followed by a 1–5 layer fully connected neural network, which ultimately produced a single output that represented the predicted value. The specific configurations can be found in the source code.

The Keras [26] (Version 2.4.3) and Tensorflow [27] (Version 2.4.1) libraries were used to perform the creation and training of the neural network predictors. Adam [28] was used as the optimizer, which internally used the mean squared error (MSE) as its loss function. A batch size of 100 images was used to train the water level predictors, while the training of the surface velocity predictors used a batch size of 15 images. The different sizes were chosen due to hardware resource restrictions. Both training methods used an initial number of 100 epochs in combination with “early stopping”, sensitive to the MSE of the validation set.

As depicted in part (I) in Figure 1, the collected footage was first split with a ratio 80:20 into sets A and the “Evaluation Set”. Set A was then split again into the following three sets: Train, Validate, Test with ratios 60:20:20. To find suitable predictors, different CNN topologies were trained on the Train and Validate sets and evaluated on Test. The relative root mean squared error (RRMSE) was used as the measure to determine the best performing network, with lower values being preferred over higher ones.

Since this was a trial-and-error phase, the Test set was used multiple times. To avoid overfitting, the best networks were ultimately evaluated once on the Evaluation Set, with the outcome presented here in the Results section.

2.6. Scenarios

One central problem of many applications involving neural networks is the availability of training data. In our case, the training data consists of images, which are labelled with the water level and the surface velocity. In practice, this data is obtained through a measurement campaign, which is time consuming, expensive, or even impossible. If training data is collected in a short period of time, it can hardly be expected that the whole range of low to high flows occur in the system. Therefore, it is important to understand how little data is needed for this approach to still provide satisfactory results.

To better understand the network performance in different situations, we defined the four following scenarios. Each scenario defines a restriction on the training set, in the form of values that are excluded. For example, this could mean that we exclude all images where the channel is filled to a degree of 50%–60% from the Train and Validation sets of the water level predictor. The Test and Evaluation Sets will then include these values and evaluate how well the network is able to interpret these unseen situations.

Scenario 1: Base case. Here, we use the complete training set as input and expect the best performing network. This would correspond to an extremely thorough measurement campaign where most of the possible scenarios are found or simulated.

Scenario 2: Two gaps. To see the influence of the gaps and their size, we define one small and one larger gap in the training data. The intention behind this scenario is to better understand the ability of the predictors to handle unseen situations, as it is very likely that the measurement campaign does not collect all possible configurations.

The data set used in the water level predictor training contained the gaps [57.5, 62.5]% and [67.5, 80]%, while the data set used to train the surface velocity predictor missed data in [57.5, 62.5] cm/s and [65, 75] cm/s.

Scenario 3: No extreme values. The smallest and biggest, i.e., the extreme values are removed from the training set. This reflects the situation that very high levels of water are unlikely to arise during the measurement period. Complete flooding of the channel might not even be feasible in practice.

The data set used in the water level predictor training contained the gaps [0, 62.5]% and [81, 100]%, while the data set used to train the surface velocity predictor missed data in [0, 62] cm/s and [73, 100] cm/s.

Scenario 4: Only extreme values. All intermediate values are cut from the training set and only both extremes are kept. While this scenario is more unlikely to arise, it is an interesting counterpart to Scenario 3 and will help to illuminate the capabilities of our predictors.

The data set used in the water level predictor training contained the gap [62, 81]%, while the data set used to train the surface velocity predictor missed data in [62, 73] cm/s.

3. Results

To provide further insight into the time series that were used in this section, we first present the measured water level and surface velocity. Figure 3 below shows the measurements in combination with their corresponding reconstructions of the CNNs of Scenario 1, i.e., under optimal data availability (a more detailed presentation of the prediction error and uncertainties will follow further below in Figures 4 and 5).

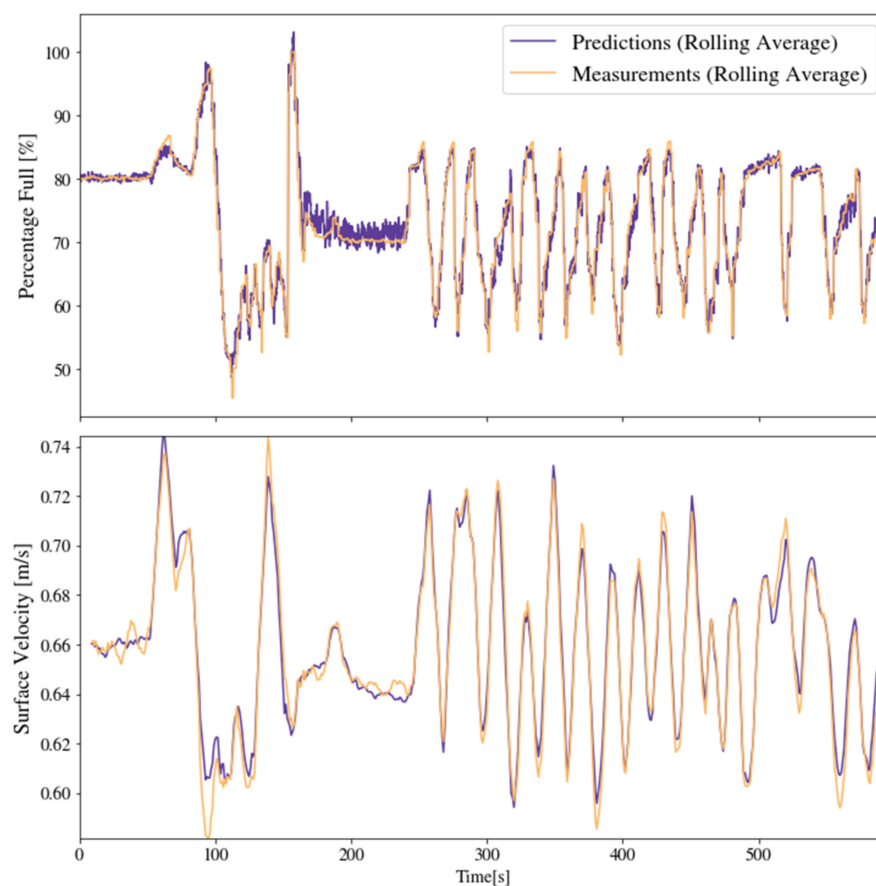


Figure 3. Water Level and Surface Velocity Measurements and Predictions Using the CNN of Scenario 1. Both the measurements as well as the predictions are smoothed using a rolling average.

As mentioned above, the water level predictions rely on single frames and thus result in 25 predictions per second, contrary to the surface velocity prediction which relies on a series of images, resulting in one prediction per second.

Throughout the remainder of this section and the next, we will rearrange the time series and sort the data points according to the y-values of the reference measurements. This allows us to clearly visualise the influence of the gaps in the training sets on the prediction accuracy.

The remainder of this section is split into the following two parts: the first part is concerned with the prediction of the water level and the second focuses on the prediction of the surface velocity. As mentioned above, the results were obtained with a test set that was evaluated once and has not been seen before while selecting the best network topology.

3.1. Water Level Prediction

The summarised results for each of the four scenarios can be seen in Figure 4. Below each scenario, the mean relative root squared error, as well as the 5th and 95th percentiles of the prediction error indicate the prediction performance. These quantile ranges demonstrate the uncertainty associated with a single prediction—smaller ranges are thus preferred. The shaded area visualises the gaps in the data set, i.e., input that has never been seen by the network during training.

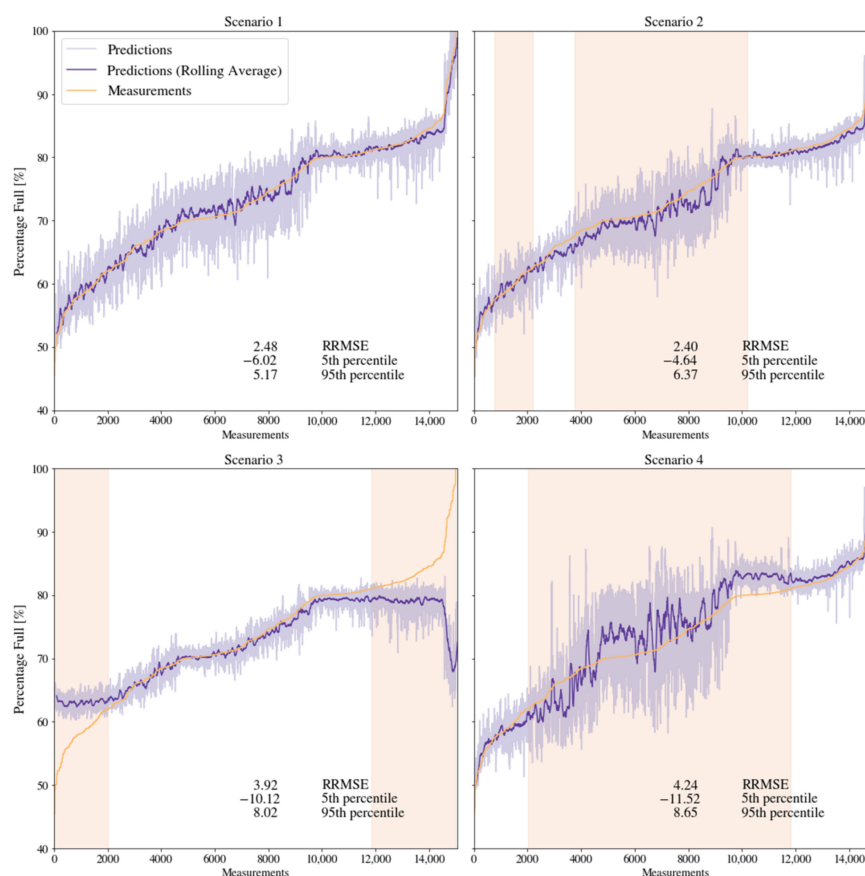


Figure 4. Water level prediction and relative errors for Scenario 1—base case, Scenario 2—two gaps, Scenario 3—no extreme values, and Scenario 4—only extreme values. Shaded areas represent missing training data, i.e., water levels that were never observed during training and need to be fully interpolated by the network. Underneath each scenario, the predictor performance is summarized with the error band of [5th percentile error, relative root mean squared error, 95th percentile error].

To clearly visualise the gaps in the training sets, the data points have been sorted according to their corresponding measured water level (values on the y-axis in Figure 4). Since there are comparatively few very high-water levels in the data set and all of them appear close together at the end of the measurement curves, a sharp increase can be observed.

We can observe from the base case of Scenario 1, that the prediction can be extremely accurate, with a RRMSE of 2.48%. The best result was achieved with a camera at position P_3 , above and to the side of the channel (see Figure 2).

Scenario 2, with only intermittent training data available, shows a similar performance compared to the base case with a RRMSE of 2.40%. We see that the network can interpret missing values reasonably well if the gap is small. As the gap grows larger, the predictor is no longer able to perform accurately, as seen with the increased error in the range $[-4.64, 6.37]\%$.

Scenario 3, with missing extreme values, clearly shows the increased error (3.92%), where the training data is missing. Contrary to Scenario 2, the predictor is not capable of interpreting the input and “flat lines” on the most extreme values it encountered in the training set.

Scenario 4, where only extreme values are given in the training set, can be seen as an extension of Scenario 2. As the gap is comparatively large, the RRMSE increases to 4.24% with a wider range of $[-11.52, 8.65]\%$.

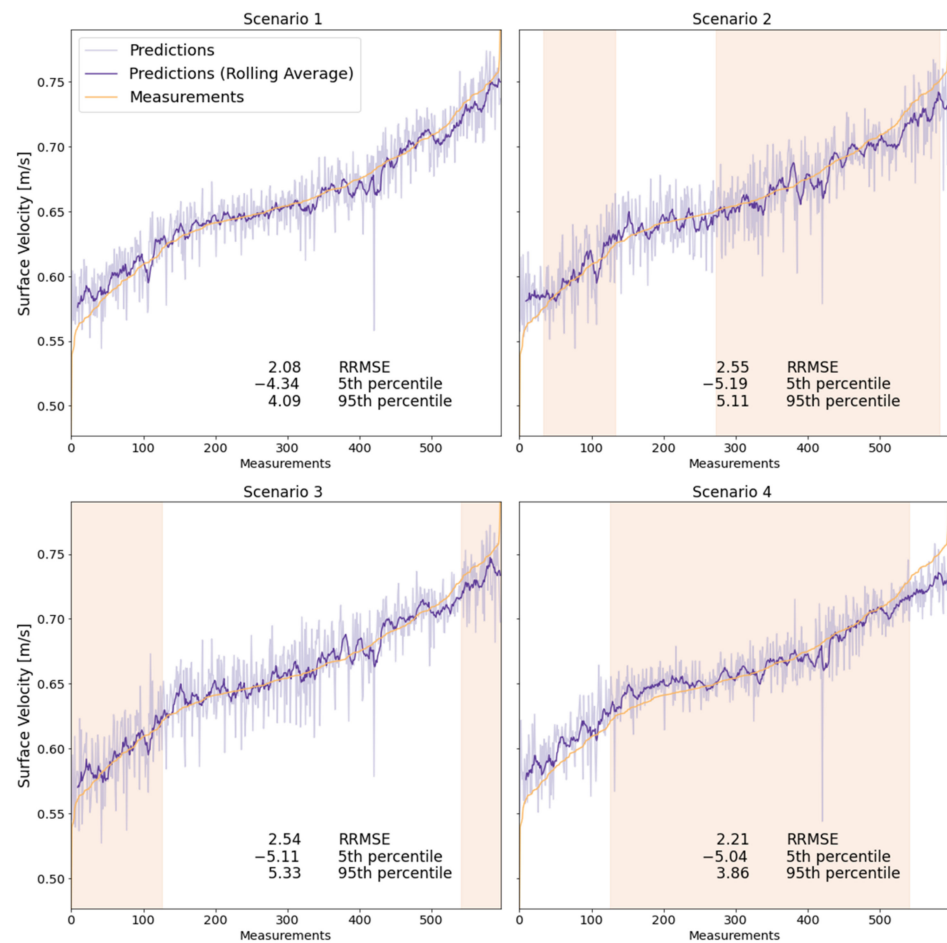


Figure 5. Surface velocity evaluation and relative errors for Scenario 1—base case, Scenario 2—two gaps, Scenario 3—no extreme values, and Scenario 4—only extreme values. Shaded areas represent missing training data, i.e., surface velocities that were never observed during training and need to be fully interpolated by the network. Underneath each scenario, the predictor performance is summarized with the error band of [5th percentile error, relative root mean squared error, 95th percentile error].

The summary of these results can be found in Table 1 below.

Table 1. Summary of Water Level Prediction Errors and Prediction Ranges [%].

Scenario	RRMSE	5th Percentile	95th Percentile
1	2.48	-6.02	5.17
2	2.40	-4.64	6.37
3	3.92	-10.12	8.02
4	4.24	-11.52	8.65

3.2. Surface Velocity Prediction

The summarised relative mean squared errors can be found in Figure 5. Similar to the behaviour of the water level prediction, we see from the base case of Scenario 1, that the prediction can be extremely accurate, with a RRMSE of 2.08% with input from position P₃.

Similar to the water level measurements shown earlier, a sharp increase towards the end of the reference measurements can be observed in Figure 5 above. Again, this is due to the smaller number of very high surface flow values in the data set, which appear close together towards the end of the curves, resulting in a sharp spike.

The performance on the individual scenarios is similar to the scenarios used in the water level prediction:

Scenario 2, with training data containing two gaps, shows the decreased performance on the parts of the data that is missing, which is to be expected. However, the algorithm does not break down and can still interpret unseen situations reasonably well with an RRMSE of 2.55% in the range [−5.19, 5.11]%.

Scenario 3, where the extreme values are cut from the data set, shows a different behaviour to the water level prediction. The network can interpret the unseen input at the edges of the dataset and does not “flat line” like the prediction of the water level. As expected, the error increases when compared to the base case to 2.54%.

Scenario 4, exhibiting a large gap in the middle of the training set increases the error further to a RRMSE of 2.21% in the range [−5.03, 3.86]%. Again, the algorithm is able to roughly estimate the velocity and does not break down completely.

Table 2 gives a summary of these results below.

Table 2. Summary of surface velocity prediction errors and prediction ranges [%].

Scenario	RRMSE	5th Percentile	95th Percentile
1	2.08	−4.34	4.09
2	2.55	−5.19	5.11
3	2.54	−5.11	5.33
4	2.21	−5.04	3.86

4. Discussion

We see in Figures 4 and 5 that the predictions can be very accurate, if training data is available and complete. In such cases, the predictions are essentially accurate reproductions of the input. This scenario is, however, highly unlikely in practice, as such a training set is most likely unavailable. A typical measurement campaign over a fixed time window is unlikely to capture the full variety that marks this base case scenario.

Through Scenarios 2–4 the consequences of missing data are explored. Smaller gaps in the training sets are manageable and the predictor can interpret unseen input. This is an important property as such gaps are unavoidable.

The problem then becomes the size of the gap and which parts of the data are missing. As expected, larger gaps lead to more inaccurate results, as seen in Scenario 4. As the range of the error increases, the confidence in the predictions naturally decreases. At a certain point, it does therefore not seem sensible to use real-valued predictions, which are particularly difficult to interpret as the network does not provide a measure of uncertainty along with its result. In such a situation, where the gap in the training set is large, it seems more sensible to use categorical results instead, which indicate a range of possible values.

A complementary problem to the size of the gap is its location. As seen in Scenario 3, missing extreme values, i.e., values at the fringes of the test set, are impossible for the network to interpret and in consequence predict correctly. The predictions tend to stay on the extremes of the training set. In other words, the networks seem to be bound to a range of possible predictions, which corresponds to the range of values encountered in the training set. Outside this range, predictions are impossible and the network “flat lines”.

This is the biggest problem of this approach in practice. As mentioned above, it is unlikely to observe these extreme situations in the time window of the measurement campaign. It is therefore necessary to artificially cause such situations. Depending on the size of the channel, however, it may be simply impractical to either completely fill or empty it, solely to obtain training footage. Complementing our approach with additional image analysis could help to bridge this gap. If artificial markers are used (similar to staff gages), the prediction could revert back to this more traditional approach, when (and if) uncertainty is detected.

As a side note, the following can be observed about surface velocity: As the water level correlates with the surface velocity, it is not clear which information is extracted from the

input images to determine the velocity. It could simply happen that the water level is taken as a proxy for the velocity and the surface structure is disregarded completely. To exclude this possibility, there have been additional tests conducted, where the border of the channel was cut from the image, to remove the information about the height. The network was still able to predict the surface velocity, albeit with an increased error. This is important, as it suggests that information about the changing surface is extracted, and the network does not solely rely on the water level to predict the surface velocity. The experiments shown above, all include the channel border again, which results in an increased performance.

4.1. Comparison

The raw hardware cost of the devices used in this experiment is less than EUR 150. As this is commodity hardware, able to run open-source software, there are no re-occurring maintenance or license fees and software fixes can be completed personally, if so needed.

Because the tests were conducted in a laboratory without sewage, there were no additional costs related to lighting, protective casing, and possibly additional data storage equipment, which would have to be included in a real-world setting. However, the cost is clearly orders of magnitude smaller compared to sensors which have to withstand direct contact with sewage (for one exemplary sensor, this would be around 100x cheaper).

Whether we are looking at the prediction of the surface velocity, or the water level, one of the main problems is the calibration data consisting of video footage and reference measurements. In situations where old equipment such as existing cameras or flow sensors are replaced, using the presented approach could be beneficial, since not both sensors need to be installed to collect calibration data. In case of both data streams existing, the calibration can even be carried out completely offline.

As with all other image-based or software-based approaches in general, the obvious advantage over dedicated hardware sensors is their ability to be improved remotely. As the algorithm is independent (to a degree) of the hardware, changes and (incremental) improvements can be deployed without interrupting the operation and even physical presence.

4.2. Water Level

Using objective reference markers such as staff gages has several advantages when it comes to the verifiability of measurements and the transferability of the underlying image analysis. Since each image contains a scale as well (i.e., the staff gage), humans can verify measurements without effort. In the presented approach, the footage is available as well and plausibility checks on measurements are possible, but as such markers are not specifically required, it can be more difficult for an operator to check the accuracy of a measurement value.

When using staff gages for water level predictions, the detection of the water edge is the core problem. Because this is such an isolated problem, in terms of the location on the image, a detection algorithm can be applied for different measurement sites, as long as the image part containing the staff gage can be isolated. In other words, the transferability of the technique is high, and a sensor can fairly easily be used at new locations. The presented approach does not have the same degree of flexibility, as a certain amount of training data must be available for each new location.

Since this is a key limiting factor, we plan to explore the transferability of our approach in future experiments. One promising possibility is to use the pre-trained kernels from the lab experiment and then use a smaller, site-specific training set for calibration [15].

The presented approach is inherently less accurate than sensors directly measuring the water level—e.g., with ultrasonic sensors—since our network effectively learns to imitate the reference sensor's behaviour (and will never be perfect). However, with a best-case error of around 2.48 %, the performance is comparable and the differences in accuracy are likely negligible in practice.

4.3. Surface Velocity

The accuracy of the surface velocity measurements under optimal conditions are comparable to existing large-scale particle image velocimetry (LSPIV) approaches, such as [20], who have an error of 1.7% compared to their reference sensors. As explained earlier, missing training data is the main problem of the presented approach, resulting in uncertainties when unseen situations are evaluated. PIV approaches have the benefit that they require only an initial calibration of the camera parameters to counteract image distortions before being operational—in every situation, as long as enough trackable particles are available. As both techniques rely on image material, both approaches have excellent cross-usage potential, as well as the possibility for manual plausibility checks.

With regard to transferability, there is the question of which input features the network actually picks up on. Since a key feature of CNNs is the ability to detect objects in an image independent of their location, we can rule out the possibility of some form of particle tracking with good certainty (as the particle location would be an important property, that should be preserved). If the network is tuned more towards the change in surface structure, which is dependent on the stream velocity and the specific channel geometry, the transferability of pre-trained kernels will most likely be lower. This behaviour needs to be explored further in a practical setting, with channels of different geometries/dimensions.

4.4. Measurement Frequency

The results presented above use an underlying prediction frequency of 1 Hz. We see that the individual predictions fluctuate around a mean value that is, naturally, more accurate. The question then arises how long a measurement window should be to produce more accurate predictions. The influence of the prediction frequency on the prediction accuracy, can be seen in Figure 6.

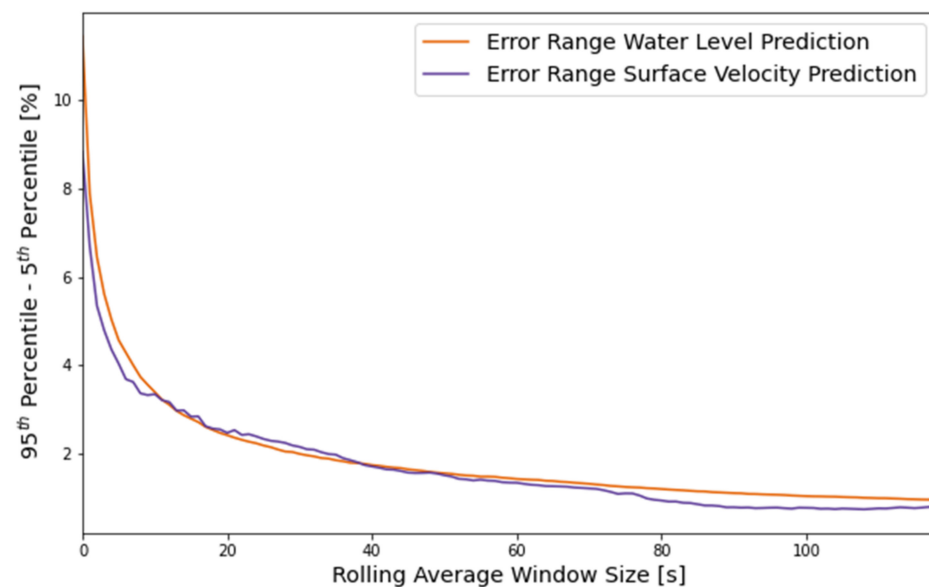


Figure 6. Influence of the prediction frequency on the error ranges of both the water level and the surface velocity predictions.

As assumed, the averaged predictions get more accurate, as more individual predictions are combined into one using a simple mean. The question is then simply which measurement frequency is needed and which error can be tolerated. This depends on the application and has no generally correct answer. A hybrid approach could be useful in many situations, where high-frequency predictions are used for short time horizons and the more accurate ones, periodically complement the values to introduce more certainty.

5. Conclusions

We presented the use of convolutional neural networks to extract the water level and surface velocity directly from camera images. Since we avoid direct contact with the raw sewage, the maintenance effort is reduced, and the sensor does not need to be as robust as it would otherwise. Using off-the-shelf hardware, which allows for open-source software to be used, reduces costs even further while at the same time preventing vendor lock-ins. Overall, the cost for each sensor used in this approach is around EUR 150 at the time of writing, making it significantly cheaper than conventional products.

The results suggest that this approach is accurate enough to compete with existing remote sensing techniques such as LSPIV, while not relying on (potentially artificial) tracer particles, as long as an adequate training set is provided. The prediction of the water level seems to be more affected by missing training data, compared to the prediction of the surface velocity; in particular, missing extreme values cannot be compensated while predicting the water level.

In terms of the overall transferability of the approach, it must be noted that, while extracting the water level from an image should not pose too much difficulty in a new environment, the transferability of the surface velocity prediction is more unclear. As it seems now, the network considers the surface structure—among other features—to derive the velocity. Since this is a channel-specific geometric property, the level of adaptation needed in a new environment—ranging from complete re-training of the CNN to simpler transfer learning—is untested at the moment.

Regardless of the degree of transferability, the method proposed here is simple to use in practice, if reference measurements are available. In a situation where old sensors are about to be phased out but are still able to provide measurements for a training set, this approach could be a very competitive option. Additionally, the proposed setup can be easily combined with other sensors to get redundancy in the measurements, while also having the added benefit of potentially being used to make more informed decisions about maintenance activities of existing sensor setups.

Author Contributions: Conceptualization, R.M., F.T.-G. and C.M.; methodology, R.M., C.M. and F.T.-G.; software, R.M.; validation, R.M., D.B.S. and F.T.-G.; formal analysis, R.M.; investigation, R.M.; resources, F.T.-G. and D.B.S.; data curation, R.M.; writing—original draft preparation, R.M.; writing—review and editing, D.B.S., F.T.-G. and C.M.; visualization, R.M.; supervision, C.M. and F.T.-G.; project administration, F.T.-G.; funding acquisition, C.M. and F.T.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All used data is archived by Zenodo (<https://doi.org/10.5281/zenodo.5833864>, accessed on 27 January 2022) and is publicly accessible. The software is available on GitHub under <https://github.com/roibert/owl>, accessed on 27 January 2022.

Acknowledgments: We would like to thank Antonio Moreno Rodenas for valuable discussions, and Thai Mai and all technicians of the NTNU hydraulics laboratory for their help in realizing the experiments and sensor construction.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Benisch, J.; Helm, B.; Bertrand-Krajewski, J.-L.; Bloem, S.; Cherqui, F.; Eichelmann, U.; Kroll, S.; Poelsma, P. Operation and maintenance. In *Metrology in Urban Drainage and Stormwater Management: Plug and Pray*; IWA Publishing: London, UK, 2021.
2. European Commission. *ATEX, Directive 2014/34/EU*; European Commission: Brussels, Belgium, 2014; Available online: <http://data.europa.eu/eli/dir/2014/34/oj> (accessed on 20 January 2022).
3. Jaafar, W.; Fischer, S.; Bekkour, K. Velocity and turbulence measurements by ultrasound pulse Doppler velocimetry. *Meas. J. Int. Meas. Confed.* **2009**, *42*, 175–182. [[CrossRef](#)]
4. Ji, H.W.; Yoo, S.S.; Lee, B.J.; Koo, D.D.; Kang, J.H. Measurement of wastewater discharge in sewer pipes using image analysis. *Water* **2020**, *12*, 1771. [[CrossRef](#)]

5. Zhang, Z.; Zhou, Y.; Liu, H.; Gao, H. In-situ water level measurement using NIR-imaging video camera. *Flow Meas. Instrum.* **2019**, *67*, 95–106. [[CrossRef](#)]
6. Mousa, M.; Oudat, E.; Claudel, C. A novel dual traffic/flash flood monitoring system using passive infrared/ultrasonic sensors. In Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), Dallas, TX, USA, 19–22 October 2015.
7. Moreno-Rodenas, A.M.; Duinmeijer, A.; Clemens, F.H.L.R. Deep-learning based monitoring of FOG layer dynamics in wastewater pumping stations. *Water Res.* **2021**, *202*, 117482. [[CrossRef](#)] [[PubMed](#)]
8. Khorchani, M.; Blanpain, O. Free surface measurement of flow over side weirs using the video monitoring concept. *Flow Meas. Instrum.* **2004**, *15*, 111–117. [[CrossRef](#)]
9. Udomsiri, S.; Iwahashi, M. Design of FIR Filter for Water Level Detection. *Eng. Technol.* **2008**, *2*, 2663–2668.
10. Kim, Y.; Muste, M.; Hauet, A.; Krajewski, W.F.; Kruger, A.; Bradley, A. Stream discharge using mobile large-scale particle image velocimetry: A proof of concept. *Water Resour. Res.* **2008**, *44*, 1–6. [[CrossRef](#)]
11. Lo, S.W.; Wu, J.H.; Lin, F.P.; Hsu, C.H. Visual sensing for urban flood monitoring. *Sensors* **2015**, *15*, 20006–20029. [[CrossRef](#)] [[PubMed](#)]
12. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. 2015. Available online: <https://arxiv.org/abs/1409.1556> (accessed on 20 January 2022).
13. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [[CrossRef](#)] [[PubMed](#)]
14. Pan, J.; Yin, Y.; Xiong, J.; Luo, W.; Gui, G.; Sari, H. Deep Learning-Based Unmanned Surveillance Systems for Observing Water Levels. *IEEE Access* **2018**, *6*, 73561–73571. [[CrossRef](#)]
15. Eltner, A.; Bressan, P.O.; Akiyama, T.; Gonçalves, W.N.; Marcato Junior, J. Using Deep Learning for Automatic Water Stage Measurements. *Water Resour. Res.* **2021**, *57*, 1–17. [[CrossRef](#)]
16. Lopez-Fuentes, L.; Rossi, C.; Skinnemoen, H. River segmentation for flood monitoring. In Proceedings of the IEEE International Conference on Big Data, Boston, MA, USA, 11–14 December 2017.
17. Jodeau, M.; Hauet, A.; Paquier, A.; Le Coz, J.; Dramais, G. Application and evaluation of LS-PIV technique for the monitoring of river surface velocities in high flow conditions. *Flow Meas. Instrum.* **2008**, *19*, 117–127. [[CrossRef](#)]
18. Muste, M.; Fujita, I.; Hauet, A. Large-scale particle image velocimetry for measurements in riverine environments. *Water Resour. Res.* **2008**, *46*, 1–14. [[CrossRef](#)]
19. Coz, J.; Jodeau, M.; Hauet, A.; Marchand, B.; Boursicaud, R. Image-based velocity and discharge measurements in field and laboratory river engineering studies using the free FUDAA-LSPIV software. In Proceedings of the International Conference on Fluvial Hydraulics: River Flow 2014, Lausanne, Switzerland, 3–5 September 2014.
20. Leitão, J.P.; Peña-Haro, S.; Lüthi, B.; Scheidegger, A.; Moy de Vitry, M. Urban overland runoff velocity measurement with consumer-grade surveillance cameras and surface structure image velocimetry. *J. Hydrol.* **2018**, *565*, 791–804. [[CrossRef](#)]
21. Jeanbourquin, D.; Sage, D.; Nguyen, L.; Schaeli, B.; Kayal, S.; Barry, D.A.; Rossi, L. Flow measurements in sewers based on image analysis: Automatic flow velocity algorithm. *Water Sci. Technol.* **2011**, *64*, 1108–1114. [[CrossRef](#)] [[PubMed](#)]
22. Benetazzo, A.; Gamba, M.; Barbariol, F. Unseeded Large Scale PIV measurements accounting for capillary—gravity waves phase speed. *Rend. Lincei* **2017**, *2*, 393–404. [[CrossRef](#)]
23. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
24. Jarrett, K.; Kavukcuoglu, K.; Ranzato, M.; LeCun, Y. What is the best multi-stage architecture for object recognition? In Proceedings of the IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009.
25. Dekking, F.M.; Kraaikamp, C.; Lopuhaä, H.P.; Meester, L.E. *A Modern Introduction to Probability and Statistics*, 1st ed.; Springer: London, UK, 2005; ISBN 978-1-84996-952-9.
26. Chollet, F. Keras, GitHub. 2015. Available online: <https://github.com/keras-team/keras> (accessed on 20 January 2022).
27. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. 2015. Available online: <https://arxiv.org/pdf/1603.04467.pdf> (accessed on 20 January 2022).
28. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. Available online: <https://arxiv.org/pdf/1412.6980.pdf> (accessed on 20 January 2022).