Is it Raining Outside?
Detection of Rainfall using General-Purpose Surveillance Cameras

Joakim Bruslund Haurum, Chris H. Bahnsen, Thomas B. Moeslund
Visual Analysis of People Laboratory
Aalborg University, Denmark
{joha, cb, tbm}@create.aau.dk

Abstract

In integrated surveillance systems based on visual cameras, the mitigation of adverse weather conditions is an active research topic. Within this field, rain removal algorithms have been developed that artificially remove rain streaks from images or video. In order to deploy such rain removal algorithms in a surveillance setting, one must detect if rain is present in the scene.

In this paper, we design a system for the detection of rainfall by the use of surveillance cameras. We reimplement the former state-of-the-art method for rain detection and compare it against a modern CNN-based method by utilizing 3D convolutions. The two methods are evaluated on our new AAU Visual Rain Dataset (VIRADA) that consists of 215 hours of general-purpose surveillance video from two traffic crossings. The results show that the proposed 3D CNN outperforms the previous state-of-the-art method by a large margin on all metrics, for both of the traffic crossings. Finally, it is shown that the choice of region-of-interest has a large influence on performance when trying to generalize the investigated methods.

The AAU VIRADA dataset and our implementation of the two rain detection algorithms are publicly available at https://bitbucket.org/aauvap/aau-virada.

1. Introduction

Varying weather and illumination conditions are a challenge for general-purpose outdoor surveillance systems [7]. In order to deal with these challenges, several image and video optimization techniques have been proposed. The purpose of these techniques is to artificially remove haze and rain from the post-processed images or video. These techniques regularize the image in order to suppress the detrimental effects of the selected weather phenomena. As a side bonus, the appearance of objects of interest in the scene should be enhanced with respect to observation by a human observer or a computer vision system.

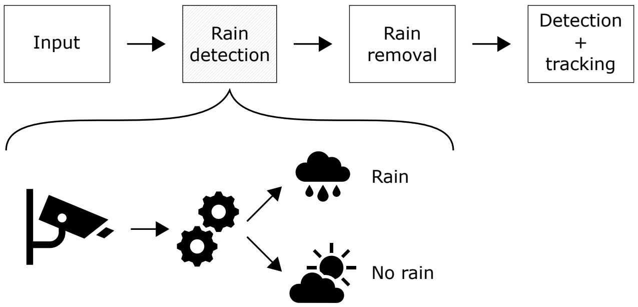


Figure 1. The proposed system at-a-glace. For rain removal algorithms to be effective in an integrated surveillance framework, the presence or absence of rain must be detected in a pre-processing step.

Because the haze and rain removal algorithms are built to improve the visibility of weather-beaten surveillance footage, they are reliant on other algorithms to detect the presence or absence of adverse weather conditions. As haze and rain removal algorithms are not general-purpose image enhancement algorithms, they will consistently deteriorate the output if either haze or rain streaks are not present in the scene. The decision process is especially important if real-time detection and tracking systems are built on top of the output of the rain removal algorithms [3, 21]. An example of such a pipeline is illustrated in Figure 1.

A prototype system for modelling the dynamic behaviour of outdoor surveillance scenes has been proposed in [2]. In this work, observations from a nearby weather station is used to guide a foreground detection algorithm. However, the applicability of the method is limited by the dependence on external weather data. It is infeasible to place a weather station alongside every surveillance camera, and the correlation of the weather data and the observations by the camera is limited if the two sensors are not in close proximity.

We would ultimately want the cameras and algorithms to be as self-reliant as possible such that they may operate without external input. A prototype of such a system was built in [16] in which the input to a foreground segmentation algorithm was pre-processed by either a rain or fog removal algorithm.

The aim of this work is to investigate the use of existing surveillance cameras as surrogate rain detectors. As opposed to existing works on detection of rain, the primary purpose of our cameras is general-purpose surveillance. This implies that we have not adjusted the camera parameters to emphasize the appearance of rain drops. Our contributions are the following:

1. A new publicly available rain dataset, the *AAU Visual Rain Dataset* (VIRADA), consisting of 215 hours of recorded video from general-purpose surveillance cameras. Ground truth measurements of rainfall are provided by a nearby laser disdrometer and a conventional tipping-bucket rain gauge.
2. A new rain detection algorithm based on the 3D CNN architecture of Tran *et al*. [33].
3. An open-source implementation of the rain level detection algorithm of Bossu *et al*. [6].
4. Evaluation of the aforementioned methods on the proposed dataset.

2. Related Work

The detection of rain streaks in images and video has been tightly coupled to the removal of the very same rain streaks since Garg and Nayar published their studies on the appearance, detection, and removal of rain in the beginning of the millenium [12, 13]. The detection of rain streaks was seen as an intermediate step in order to suppress the streaks in the final, rain-removed image or video.

Garg and Nayar noted that rain streaks appear brighter than their background and that the fast motion of the streaks imply that each streak is only visible in a single frame. Combined with the assumption of a quasi-static background, they detected rain streaks by using the photometric constraint:

$$\Delta I = I_n - I_{n-1} = I_n + I_{n+1} \geq c \qquad (1)$$

where c is a threshold and I_n denotes the image at frame n. Because the candidate streak image ΔI contains many false positives, different post-processing steps are required to filter the candidates. The photometric constraint of Equation 1 is commonly used for the initial segmentation of rain streaks in many video-based rain removal algorithms [6, 23, 31, 35].

Other approaches for detecting rain in the image space include morphological component analysis [19] or matrix decomposition [20]. These methods may be applied either on single images [9, 19] or video streams [18, 20, 30].

Recently, the popularity of convolutional neural networks (CNNs) has reached the rain removal community. Amongst those works, some architectures explicitly produce a rain image that is used to refine the rain removal process [22, 37]. These methods might be applicable for stand-alone rain detection if their training process is tuned to the estimation of rain density and not the restoration of the rain-removed image.

A different approach for detecting rain in a video was proposed by Barnum *et al.* [4] who noted that the directional uniformity of the rain streaks was ideal for detection in the frequency space. They thus transferred the image into the Fourier domain where rain streaks were lying along an elongated ellipsis. However, the authors did not investigate if the volume of rain can be estimated in the frequency space.

For readers interested in a detailed overview of rain removal algorithms, we refer to a dedicated survey [3].

2.1. Rain Density Estimation

Bossu *et al.* [6] pioneered in using the detected rain streaks as a surrogate rain gauge. Motivated by the photometric constraint of Equation 1, they used a Mixture of Gaussians (MoG) [32] to model foreground and background objects. The candidate streaks were found by the following rule:

$$\Delta I = I_{FG} - I_{BG} \geq c \qquad (2)$$

where I_{FG} and I_{BG} are the foreground and background images of the MoG model, respectively. False positives in ΔI are suppressed based on their size. The rotation of the remaining streaks are used to construct a Histogram of Orientation of Streaks (HOS). The appearance of the histogram is modelled using a Gaussian-uniform distribution, and the relationship between the Gaussian and the uniform parts of the distribution is used to detect the presence or absence of rain.

The work of Allamano *et al*. [1] utilizes rigorous formulations of camera geometry to estimate the real-world volume of the detected rain drops. The photometric constraint is used to segment candidate streaks. The width and height of these streaks together with the focal length of the camera are used to estimate the rain rate.

The subsequent work of Dong *et al*. [11] filters the candidate streaks by orientation and discards streaks not within the dominant orientation. Focused and unfocused streaks are distinguished based on intensity and edge information and used for estimating the length of each streak. The rain rate is estimated from the streak diameters by using a Gamma distribution.

Recent work of Jiang *et al*. [17] uses matrix decomposition to segment rain streaks from the background. The width of the detected streaks and the number of streaks are used to infer the rain rate. The authors use a Gamma distribution similar to [11].

The rain detection algorithm of Bossu *et al*. [6] is evaluated on footage from a general-purpose surveillance camera whereas the approaches of [1, 11, 17] are evaluated on footage from videos cameras whose parameters are tuned with the sole purpose of emphasizing the visual appearance of rain streaks.

3. The AAU Visual Rain Dataset

To the best of our knowledge, there is currently no publicly available dataset for benchmarking the detection of rain with general-purpose surveillance cameras. In order to fill this gap, we present the new publicly available *AAU Visual Rain Dataset* (VIRADA) that contains a total of 215 hours of surveillance video from two different locations in Aalborg, Denmark. The cameras are configured and positioned for traffic surveillance applications and not specifically configured for the task of detecting rain.

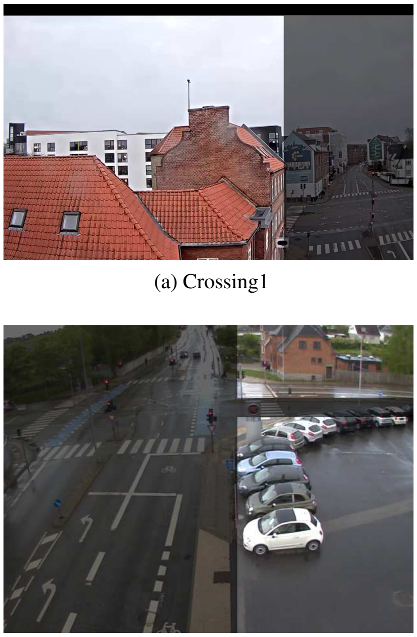
We obtain ground-truth precipitation data from two different rain gauges: a traditional, mechanical tipping-bucket rain gauge and a more advanced laser disdrometer [24]. The two measurement devices are explained in the following.

3.1. Rain Measurement Devices

Tipping-bucket rain gauge In the tipping-bucket rain gauge, rain drops are collected by a funnel that channels the water into one of two seesaw-like buckets. When a bucket is full, it dumps the water and leaves the collection of water to the second bucket. An electric signal is generated whenever a container is full and the water is dumped. This type of measurement device is widely used and large networks of the devices have been utilized for different engineering domains [25, 26].

The resolution of the buckets is 0.2 mm which means that the bucket only tips once 0.2 mm of rain has passed trough the funnel. This implies that for low-intensity rainfall, *e.g.* 0.1 - 2 mm/hour, it might take several minutes or even hours for the bucket to tip and for rain to be detected [14]. The signals generated by the tipping scales are post-processed in order to generate per-minute estimates of the precipitation level.

Laser disdrometer The laser disdrometer is an optical sensor that is capable of detecting single rain drops [24]. A laser transmitter that transmits a sheet of light in free-air is located at the left side of the device. The sheet of light is detected on the right side of the device by an optical receiver. Because the laser disdrometer is capable of detecting individual rain drops, the temporal resolution is superior compared to the mechanical tipping-bucket rain gauge. Therefore, the laser disdrometer may be used for ground truth measurements when validating radar precipitation estimates [27, 28].


(a) Crossing1

(b) Crossing2

Figure 2. Sample views of the traffic crossings from the AAU VIRADA dataset. Discarded regions are shown with a semi-transparent overlay. We denote the upper region of Crossing2 as Crossing2-brick whereas the lower region is denoted as Crossing2-asphalt.

3.2. Video Surveillance Sequences

We have collected video footage from two different traffic crossings, in the following denoted as Crossing1 and Crossing2. The Crossing2 sequence is recorded in 2013 using an AXIS M114-E camera whereas the Crossing1 sequence is recorded in 2018 using a newer AXIS Q1615-E camera. Sample footage from the two crossings are shown in Figure 2. In order to ease the task at hand, we only consider regions in the video with few moving objects due to the following reasons:

1. The detection of rain from general-purpose surveillance cameras is hard. In order to solve the problem, we should first solve the simpler sub-problem.

2. For many surveillance applications, it is possible to select a region of interest where objects are mostly static, for instance the facade of a building.

3. The selection of a region with no moving road users allows the public release of the dataset.

Dataset	Duration (hh:mm)	Frames	Frame rate	Native resolution	Cropped resolution	Camera model	Distance to gauge	
							Mech.	Laser
Crossing1-trn	87:38	9,276,654	30	1024 × 640	700 × 612	AXIS Q1615-E	580 m	1230 m
Crossing1-val	20:37	2,184,499	30	1024 × 640	700 × 612	AXIS Q1615-E	580 m	1230 m
Crossing2-tst	106:59	9,463,287	25	640 × 480	276 × 338, 276 × 112	AXIS M114-E	1820 m	970 m

Table 1. Overview of the AAU VIRADA dataset. Mech. denotes the mechanical tipping-scale rain gauge whereas Laser denotes the laser disdrometer. The two noted cropped resolutions for the Crossing2-tst dataset are for the asphalt and brick crops, respectively.

	Detected rain %	
Measurement device	Laser	Mech.
Crossing1-trn	19.67	17.97
Crossing1-val	20.86	14.63
Crossing2-tst	8.68	1.65

Table 2. Overview of the ratio of detected rain for the AAU VIRADA dataset, per measurement device. Mech. denotes the mechanical tipping-scale rain gauge whereas Laser denotes the laser disdrometer.

The discarded regions are masked out in Figure 2 with a semi-transparent overlay. A single region is chosen for Crossing1, whereas Crossing2 is split into two regions, Crossing2-brick and Crossing2-asphalt. Due to privacy concerns, only the parts of Crossing2-asphalt with no pedestrians are publicly available.

The footage from Crossing1 is split into a training (trn) and a validation (val) set whereas the Crossing2 is used in its entirety for testing. An overview of the dataset and the distance from the cameras to the dedicated rain measurement devices is found in Table 1. The ratio of rain detected by the rain measurement devices is listed in Table 2.

4. Methods

We evaluate two different methods for detecting rain from surveillance video: Bossu *et al*. rain detection [6] and the C3D CNN architecture by Tran *et al*. [33], each representing a different paradigm. The rain-detection algorithm by Bossu *et al*. represents a hand-crafted algorithm specifically designed for this task. On the other hand, the C3D CNN was originally developed for action recognition, scene classification and object recognition where temporal information is encoded through 3D network layers.

4.1. Bossu Rain Detection

Bossu *et al*. [6] propose a rain detection algorithm which in its core is based on detecting the approximate angle of the rain streaks in an image. This is done by assuming the rain streaks to be Gaussian distributed around a center angle θ with uncertainty $d\theta$. Based on the estimated distribution parameters, it can be decided whether rain is present or not. The algorithm is illustrated in Figure 3.

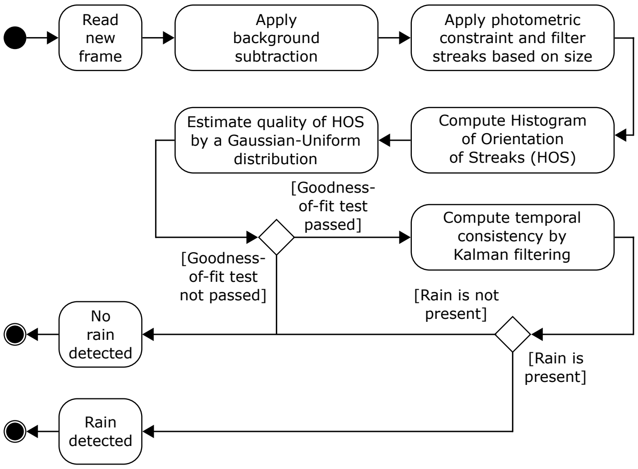

Read new frame
Apply background subtraction
Apply photometric constraint and filter streaks based on size
Compute Histogram of Orientation of Streaks (HOS)
Estimate quality of HOS by a Gaussian-Uniform distribution
[Goodness-of-fit test passed]
[Goodness-of-fit test not passed]
Compute temporal consistency by Kalman filtering
[Rain is not present]
[Rain is present]
No rain detected
Rain detected

Figure 3. Activity diagram of the rain detection algorithm by Bossu *et al.* [6].

Background subtraction and computation of candidate streaks are described in Section 2.1 and Equation 2. In the following, the remaining steps of the algorithm are described. We refer the reader to the original article [6] for a complete reference.

Histogram of Orientation of Streaks In order to determine the general orientation of the rain streaks, we create a 180-bin histogram in the range [0, 179]. We approximate the rain streak BLOBs as ellipses. To determine the orientation of the ellipses, we compute the geometric moments of the ith BLOB based on the central second-order moments, m_i^{20}, m_i^{11}, and, m_i^{02}. This leads to the calculation of θ_i, $d\theta_i$, and, w_i as shown in Equation 3-5. dm is an empirically chosen scaling constant of the uncertainty, and λ_i^1 is the largest eigenvalue of the matrix $\begin{bmatrix} m_i^{20} & m_i^{11} \\ m_i^{11} & m_i^{02} \end{bmatrix}$.

$$\theta_i = \frac{1}{2}\tan^{-1}(\frac{2m_i^{11}}{m_i^{02} - m_i^{20}}) \tag{3}$$

$$d\theta_i = \frac{\sqrt{(m_i^{02} - m_i^{20})^2 + 2(m_i^{11})^2}}{(m_i^{02} - m_i^{20})^2 + 4(m_i^{11})^2} dm \tag{4}$$

$$w_i = \sqrt{\lambda_i^1} \tag{5}$$

The Histogram of Orientation of Streaks (HOS) can then be computed, where B is the total amount of BLOBs:

$$h(\theta) = \sum_{i}^{B} \frac{w_i}{d\theta_i\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{\theta-\theta_i}{d\theta_i})^2} \qquad (6)$$

HOS quality estimation The HOS is built on the assumption that all BLOBs in the image are representations of actual rain streaks and that the orientation of the rain streaks follows a Gaussian distributed. BLOBs that represent noise in the foreground segmentation or stem from other non-rain scene elements thus have to be removed. We assume that non-rain BLOBs in the image are uniformly distributed which means that the HOS can be modeled by a Gaussian-uniform distribution:

$$y(\theta) \sim \Pi\,\mathcal{N}(\theta|\mu,\sigma) + (1-\Pi)\mathcal{U}_{[0,179]}(\theta) \qquad (7)$$

where Π denotes the ratio of the Gaussian distribution in the HOS. We assume that the rain streaks contribute to the Gaussian part of the distribution.

The parameters μ, σ and Π are estimated through an Expectation-Maximization (EM) algorithm based on the computed HOS in Equation 6.

When the EM algorithm has converged, the final HOS has to be evaluated. If the EM determined parameters does not result in a distribution that is close to the actual observed histogram $h(\theta)$, then the frame is discarded from further processing. In order to quantify this statement, a Kolomogrov-Smirnov goodness-of-fit test is performed:

$$D = \sup_{\theta} |F_n(\theta) - F(\theta)| \qquad (8)$$

where $F(\theta)$ is the cumulative distribution function of a Gaussian with the EM-estimated parameters and $F_n(\theta)$ is the accumulated HOS histogram $h(\theta)$, an empirical cumulative distribution function.

If D is above some threshold D_c, it is determined that it is not raining in the frame. If $D \leq D_c$, we estimate the temporal consistency in the following.

Temporal consistency In order to be temporally consistent in the detection of rain, a Kalman filter is used to track and smooth the EM estimated parameters. If the estimated Gaussian ratio, Π, is larger than some threshold Π_{rain}, the Kalman filter is updated and rain is detected.

4.2. C3D Convolutional Neural Network

The use of 3D convolutions to encode temporal information has been an ongoing topic in the research community since Tran *et al*. [33] proposed their C3D architecture. Some of the recent advances include more complex networks [36], residual networks [15], and separable convolutional layers [34]. However, in order to provide a baseline for video-based rain detection, we investigate the well-established original C3D network.

The C3D CNN architecture builds on the concept of using series of consecutive video frames as input and utilizing 3D convolutions instead of 2D convolutions. Specifically, each input is changed from a 3D tensor of size $c \times h \times w$ to a 4D tensor of size $c \times l \times h \times w$. The parameters c, w, and h are the number of channels and the height and width of the the input whereas l is the length of the input sequence. The receptive field of the filters is also changed from $k \times k$ to $d \times k \times k$, where k and d are the spatial and temporal extent of the filter, respectively.

The original C3D network ends with two fully-connected layers of size 4096, a dropout rate of 50%, and a softmax layer. This has the disadvantage of forcing a specific input size for the image, meaning the input should be cropped or resized. In order to get a single output for the entire image, we convert the network to a fully-convolutional network (FCN) by replacing the fully-connected layers with 2D convolutional layers and adding a global averaging layer on top of the softmax layer. The two new convolutional layers, Conv6a and Conv6b, have filter sizes of $512 \times 4 \times 4$ and $4096 \times 1 \times 1$ in order to function in a similar way as a fully-connected layer. The modified network architecture is shown in Figure 4. By converting the network to a FCN, it effectively applies a sliding window approach with a spatial stride of 32.

5. Implementation

In the following, we will guide the reader through the most important implementation details of the investigated methods. Due to the inherently higher precision of the laser disdrometer, as discussed in Section 3, only labels from the laser disdrometer are utilized for training and evaluation in this work. Our implementation is publicly available.

5.1. Bossu

As the method from Bossu *et al*. [6] was not publicly available, we have implemented it from scratch in C++ with the OpenCV framework. We use the first 500 frames of each video to initialize the background model of MoG. The EM algorithm is initialized according to the instructions of Bossu *et al*. [5]. The process and measurement noise covariance matrices of the Kalman filter are initialized with a variance of 0.01 and 0.1, respectively, and a covariance of 0, as per the original authors [6].

In order to determine the remaining parameter values, we perform a parameter search on six video snippets from the Crossing1 dataset with an equal amount of rain and non-rain videos. The duration of the snippets vary from 5 to 20 minutes. A total of 9600 parameter combinations are investigated and the specific values included in the search are shown in Table 3. We selected the final parameters based on the following criteria:

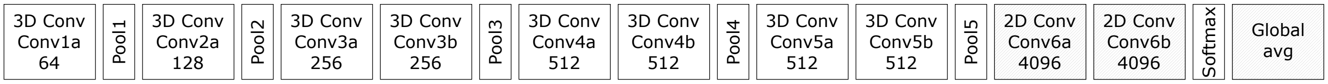


Figure 4. Overview of the modified C3D CNN architecture. 3D conv and 2D conv denotes 3D and conventional 2D convolutions, respectively. Pool denotes max pooling and the number of filters are denoted in the bottom of each box. Our modifications of the network are marked in grey. Figure adapted from [33].

Parameter	Search space	Selected value
MoG warm-up frames	[500]	500
c	[3, 5]	3
Minimum BLOB size	[4]	4
Maximum BLOB size	[50:50:200]	200
dm	[0.5:0.5:2.0]	0.50
EM max iterations	100	100
D_c	[0.01:0.01:0.20]	0.19
Π_{rain}	[0.20:0.02:0.50]	0.40

Table 3. Values and search space for the Bossu parameter search. The ranges in the search space follow the python convention, with [3,5] being a list of parameters and [0.5:0.5:2.0] referring to values in the range from 0.5 to 2.0 with an interval of 0.5

- For rain sequences, the Bossu algorithm should detect rain for at least 60 % of the frames, preferably more.
- For no-rain sequences, the Bossu algorithm should detect rain for maximum 40 % of the frames, preferably less.

The collection of parameters that performs most consistently under these criteria is listed in the rightmost column of Table 3.

5.2. C3D

We train the C3D network from scratch on the training/validation split of the Crossing1 videos, utilizing 2 Tesla V100 graphic cards. The network is trained as a binary classification problem on 112 × 112 sized crops from the videos in order to maintain a reasonable batch size. We load video sequences with a temporal stride of 8 frames.

We use a stochastic gradient descent optimizer with momentum, weight decay, and a step-based learning rate scheduler for training. The learning rate scheduler multiplies the learning rate by γ every s epochs. The set of hyperparameters used during training are listed in Table 4. The PyTorch deep learning framework [29] is used for creating and training the model and the sequence loading was handled using the NVIDIA Video Loader framework [8]. We augment the data by randomly chosen crops and random flipping along the vertical axis with 50% chance.

Validation For the Crossing1 videos, the FCN structure results in a 19 × 17 patch grid being investigated while for the Crossing2-tst videos, a 6 × 8 and 6 × 1 patch grid is investigated for the asphalt and brick crops, respectively. Each patch outputs a vector containing the output of the softmax layer. These vectors are subsequently averaged and thresholded in order to get the final binary prediction for the video sequence.

Hyperparameter	Value
Batch size	128
Sequence stride	8
Learning rate	0.01
Momentum	0.9
Weight decay	0.0001
γ	0.1
s	5

Table 4. C3D hyperparameters.

The network is trained for 57 epochs and reached a training accuracy of 94.03% and a validation accuracy of 87.38%. The accuracy and loss plots are shown in Figure 5 and Figure 6.

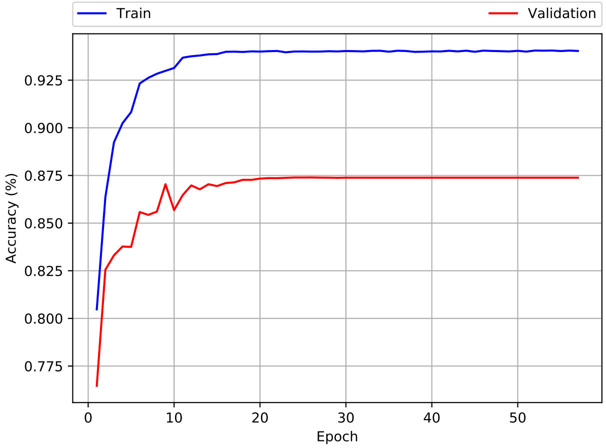


Figure 5. Average accuracy per epoch for the trained C3D CNN.

6. Experimental Results

The trained algorithms are evaluated on the AAU VIRADA dataset presented in Section 3. In order to get a quantitative measure of the rain detection performance, several metrics are used. First, the values of the confusion matrix are reported: True Positive (TP), True Negative (TN),

Sequence	Method	TP	TN	FP	FN	Acc	F1	MCC
Crossing1-trn	C3D-FCN	215244	923596	7766	12802	**0.9823**	**0.9544**	**0.9435**
	C3D-Center	202932	920692	10672	25114	0.9691	0.9190	0.9007
	Bossu-EM	1096369	3498967	3915541	719777	0.4978	0.3211	0.0603
	Bossu-Kalman	1119361	3460093	3954415	696785	0.4961	0.3249	0.0663
Crossing1-val	C3D-FCN	30126	208447	7405	27042	**0.8738**	**0.6362**	**0.5821**
	C3D-Center	25320	199555	16298	31847	0.8237	0.5126	0.4159
	Bossu-EM	263008	912983	805113	192395	0.5411	0.3453	0.0887
	Bossu-Kalman	267253	909237	808859	188150	0.5413	0.3490	0.0945
Crossing2-asphalt	C3D-FCN	0	1069231	0	102717	**0.9124**	0.0000	0.0000
	C3D-Center	245	1039578	40409	102474	0.8792	0.0034	-0.0837
	Bossu-EM	224853	6335804	2257136	591994	0.6972	0.1363	**0.0080**
	Bossu-Kalman	234181	6264711	2328229	582666	0.6907	**0.1386**	0.0010
Crossing2-brick	C3D-FCN	72619	729561	350381	30095	**0.6783**	0.2763	0.2248
	C3D-Center	75690	720369	359557	27024	0.6731	**0.2814**	**0.2359**
	Bossu-EM	281084	5837499	2755441	535763	0.6502	0.1459	0.0141
	Bossu-Kalman	290583	5762519	2830421	526264	0.6433	0.1476	0.0158

Table 5. Rain detection results on the AAU VIRADA dataset, using labels from the laser disdrometer.

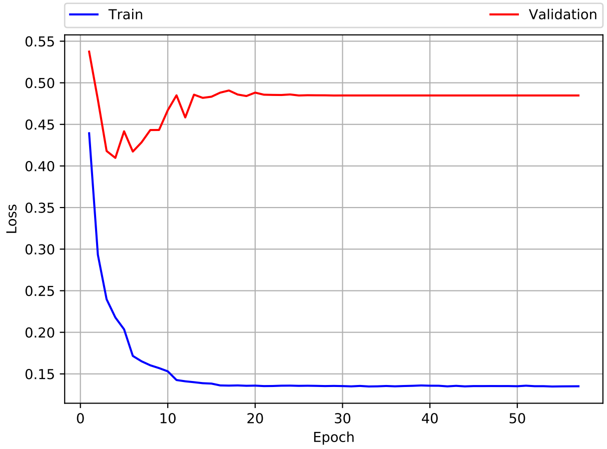


Figure 6. Average loss per epoch for the trained C3D CNN.

False Positive (FP), and False Negative (FN). In this case a True Positive is when rain is correctly detected, while a True Negative is when no rain is correctly detected. Based on these quantities, the **Accuracy** (Acc), **F1 Score** (F1), and **Matthews Correlation Coefficient** (MCC) are calculated as follows:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{9}$$

$$\text{F1} = \frac{2\ \text{TP}}{2\ \text{TP} + \text{FP} + \text{FN}} \tag{10}$$

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{11}$$

The accuracy metric is an often used metric which directly indicates the correct percentage of assigned frames. However, it is not a good indicator for imbalanced datasets and can in these cases be misleading. The F1 score and MCC try to counteract this problem.

The F1 score provides a metric where true negatives are not considered, which for datasets skewed towards a high amount of trivial true negatives results in a more fair representation.

In the same sense, the MCC metric tries to provide a fair single value representation of the confusion matrix, even for imbalanced datasets, by providing a value in the range [-1; 1]. -1 indicates total disagreement, 0 indicates pure guesswork, and 1 indicates perfect predictions. If any of the sums in the denominator results in 0, the resulting value is set to 0. The MCC metric is a measure which have been recommended for computational biology and biomedical and health informatics due to its built-in considerations for both positive and negative predictions in imbalanced datasets [10]. MCC will be used as the primary evaluation metric.

The method evaluation is not a one-to-one comparison, as the Bossu rain detection algorithm works on a per-frame basis while the C3D CNN analyse 16 frames at a time, with a 8 frame stride. Therefore, there will be fewer predictions for the C3D CNN. In order to demonstrate the effect of the FCN structure, the C3D CNN will be evaluated when applied on the entire frame, utilizing the FCN structure, and evaluated with a 112 × 112 center patch of the frame. These are denoted *C3D-FCN* and *C3D-Center*, respectively. Furthermore, the Bossu algorithm is evaluated in two ways, by investigating the rain detection capabilities when using either the per-frame EM estimated HOS parameters, or the Kalman smoothed HOS parameters. These are denoted

Bossu-EM and *Bossu-Kalman*, respectively. The laser disdrometer labels have been converted from per-minute to per-frame representations. The results are shown in Table 5. The best performing metrics are highlighted in bold.

From the results it is evident that the C3D CNN outperforms the Bossu algorithm on all of the Crossing1 videos. The Bossu rain detector algorithm provides nothing more than a random guess, as shown by the accuracy values near 50% and MCC values near 0. On the other hand, the modified C3D CNN achieves a near perfect accuracy of 98 % and MCC of 0.94 on the training set whereas a 87% accuracy and MCC of 0.58 is scored on the validation set. This indicates that while it performs well, performance can be improved, as shown by the large amount of false negative predictions. Comparatively, if only the center is evaluated with the C3D CNN, the performance drops drastically to a MCC of 0.90 and 0.41 on the training and validation sets, respectively. As we trained the C3D CNN on a subset of the Crossing1 dataset and determined the parameters of the Bossu algorithm on the very same dataset, the difference in performance is striking.

On the Crossing2 videos, two crops are tested: One with asphalt background and one with a brick house background. It is shown that neither the C3D CNN nor the Bossu algorithm generalize well when tested on the asphalt background. The C3D CNN evaluating the entire frame predicts no rain, while the Bossu algorithm predicts rain approximately one third of the time. The C3D CNN evaluated on just the center patch does predict rain in some instances, but due to the large discrepancy between true and false predictions, it results in a MCC of -0.08. When tested on the brick house background, however, the C3D CNN outperforms the Bossu rain detector on all metrics. This indicates that the C3D CNN can generalize somewhat when evaluated on surfaces similar to the one it was trained on. It is also found that by just evaluating the center patch with the C3D CNN, the MCC increases by 0.01.

The results also show that the Bossu algorithm works better on the brick house background but that the performance is still affected by a large amount of false positives and negatives.

We hypothesize that the reason C3D-Center performs better than C3D-FCN on the Crossing2 data, is due to the dynamic effects that occurs in the regions. In Crossing2-asphalt there are many cars with reflections, while in Crossing2-Brick there are pedestrians walking by along the sidewalk. By using just the center patch, some of these dynamic effects may be avoided. Further investigation is needed in order to be certain.

7. Conclusion

In this work we investigated detection algorithms for general-purpose surveillance cameras. The current state-of-the-art method and a data-driven 3D CNN method were implemented and compared on a new publicly available dataset, the AAU Visual Rain Dataset (VIRADA), consisting of 215 hours from two separate traffic crossings, making it by far the biggest rain dataset captured by general purpose surveillance cameras. A subset of one of the traffic crossing videos was used to train the algorithms. When testing on unseen data from the traffic crossing the algorithms were trained on, we found that our modified 3D CNN algorithm outperformed the previous state-of-the-art method. However, when testing on data from a new traffic crossing, the performance of the algorithms were dependent on the similarity of the investigated region-of-interest and the training data. Using a similarly textured region-of-interest, our 3D CNN outperformed the previous state-of-the-art by a large margin. Comparatively, when using a region-of-interest with a very different kind of texture, our 3D CNN failed to function. From these observations it is clear that our modified 3D CNN outperforms the previous state-of-the-art, but also that the task of rain detection for general-purpose surveillance cameras is not yet solved.

Future work could include an in-depth comparison between the laser disdrometer and the mechanical tipping-scale rain gauge in order to determine the effect of the label quality on the evaluated results.

.

References

[1] P Allamano, A Croci, and F Laio. Toward the camera rain gauge. *Water Resources Research*, 51(3):1744–1757, 2015. 2, 3

[2] Thiemo Alldieck, Chris Bahnsen, and Thomas Moeslund. Context-aware fusion of rgb and thermal imagery for traffic monitoring. *Sensors*, 16(11):1947, 2016. 1

[3] Chris H Bahnsen and Thomas B Moeslund. Rain removal in traffic surveillance: Does it matter? *IEEE Transactions on Intelligent Transportation Systems*, 2018. 1, 2

[4] Peter C Barnum, Srinivasa Narasimhan, and Takeo Kanade. Analysis of rain and snow in frequency space. *International journal of computer vision*, 86(2-3):256, 2010. 2

[5] Jérémie Bossu, Nicolas Hautiere, and Jean-Philippe TAREL. Utilisation d'un modèle probabiliste d'orientation de segments pour détecter des hydrométéores dans des séquences vidéo. In *XXIIe colloque GRETSI (traitement du signal et des images), Dijon (FRA), 8-11 septembre 2009*. GRETSI, Groupe dEtudes du Traitement du Signal et des Images, 2009. 5

[6] Jérémie Bossu, Nicolas Hautière, and Jean-Philippe Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *International Journal of Computer Vision*, 93(3):348–367, Jul 2011. 2, 3, 4, 5

[7] Norbert Buch, Sergio A Velastin, and James Orwell. A review of computer vision techniques for the analysis of urban traffic. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):920–939, 2011. 1

[8] Jared Casper, Jon Barker, and Bryan Catanzaro. Nvvl: Nvidia video loader. `https://github.com/NVIDIA/nvvl`, 2018. 6

[9] Bo-Hao Chen, Shih-Chia Huang, and Sy-Yen Kuo. Error-optimized sparse representation for single image rain removal. *Industrial Electronics, IEEE Transactions on*, 64(8):6573–6581, 2017. 2

[10] Davide Chicco. Ten quick tips for machine learning in computational biology. *BioData Mining*, 10(1):35, Dec 2017. 7

[11] Rong Dong, Juan Liao, Bo Li, Huiyu Zhou, and Danny Crookes. Measurements of rainfall rates from videos. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–9. IEEE, 2017. 2, 3

[12] Kshitiz Garg and Shree K Nayar. Photometric model of a rain drop. In *CMU Technical Report*. 2003. 2

[13] Kshitiz Garg and Shree K. Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, Oct. 2007. 2

[14] Emad Habib, Witold F Krajewski, and Anton Kruger. Sampling errors of tipping-bucket rain gauge measurements. *Journal of Hydrologic Engineering*, 6(2):159–166, 2001. 3

[15] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6546–6555, 2018. 5

[16] Nicolas Hautiere, Erwan Bigorgne, Jérémie Bossu, and Didier Aubert. Meteorological conditions processing for vision-based traffic monitoring. In *The Eighth International Workshop on Visual Surveillance-VS2008*, 2008. 2

[17] Shijie Jiang, Vladan Babovic, Yi Zheng, and Jianzhi Xiong. Advancing opportunistic sensing in hydrology: a novel approach to measuring rainfall with ordinary surveillance cameras. *Water Resources Research*, 2019. 3

[18] Tai-Xiang Jiang, Ting-Zhu Huang, Xi-Le Zhao, Liang-Jian Deng, and Yao Wang. A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors. In *Computer Vision and Pattern Recognition, IEEE Conference on*, 2017. 2

[19] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 21(4):1742–1755, 2012. 2

[20] Jin-Hwan Kim, Jae-Young Sim, and Chang-Su Kim. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE Transactions on Image Processing*, 24(9):2658–2670, 2015. 2

[21] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. *arXiv preprint arXiv:1903.08558*, 2019. 1

[22] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3233–3242, 2018. 2

[23] Peng Liu, Jing Xu, Jiafeng Liu, and Xianglong Tang. Pixel based temporal analysis using chromatic property for removing rain from videos. *Computer and Information Science*, 2(1):53, 2009. 2

[24] Martin Löffler-Mang and Jürg Joss. An optical disdrometer for measuring size and velocity of hydrometeors. *Journal of Atmospheric and Oceanic Technology*, 17(2):130–139, 2000. 3

[25] H. Madsen, P.S. Mikkelsen, D. Rosbjerg, and P. Harremos. Estimation of regional intensity-duration-frequency curves for extreme precipitation. *Water Science and Technology*, 37(11):29 – 36, 1998. Use of Historical Rainfall Series for Hydrological Modelling. 3

[26] P.S. Mikkelsen, H. Madsen, K. Arnbjerg-Nielsen, H.K. Jrgensen, D. Rosbjerg, and P. Harremos. A rationale for using local and regional point rainfall data for design and analysis of urban storm drainage systems. *Water Science and Technology*, 37(11):7 – 14, 1998. Use of Historical Rainfall Series for Hydrological Modelling. 3

[27] Jesper Ellerbæk Nielsen, Søren Liedtke Thorndahl, and Michael R. Rasmussen. Improving weather radar precipitation estimates by combining two types of radars. *Atmospheric Research*, 139(March):3645, 2014. 3

[28] Jesper Ellerbæk Nielsen, Søren Liedtke Thorndahl, and Michael R. Rasmussen. A numerical method to generate high temporal resolution precipitation time series by combining weather radar measurements with a nowcast model. *Atmospheric Research*, 138(March):1–12, 2014. 3

[29] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 6

[30] Weihong Ren, Jiandong Tian, Zhi Han, Antoni Chan, and Yandong Tang. Video desnowing and deraining based on matrix decomposition. In *Computer Vision and Pattern Recognition, IEEE Conference on*, July 2017. 2

[31] Varun Santhaseelan and Vijayan K Asari. Utilizing local phase information to remove rain from video. *International Journal of Computer Vision*, 112(1):71–89, 2015. 2

[32] Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):747–757, 2000. 2

[33] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 4489–4497, Washington, DC, USA, 2015. IEEE Computer Society. 2, 4, 5, 6

[34] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *2018 IEEE/CVF Conference on Com-*

puter Vision and Pattern Recognition, pages 6450–6459, June 2018. 5

[35] AK Tripathi and S Mukhopadhyay. Video post processing: low-latency spatiotemporal approach for detection and removal of rain. *IET Image Processing*, 6(2):181–196, 2012. 2

[36] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *CVPR*, 2018. 5

[37] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1357–1366, 2017. 2