Deep Hazardous Events Detection in Top-Down Fish-Eye Images for Railway Applications

Olivier Laurendin, Anthony Fleury, Sébastien Ambellouis, Sanaa Chafik, Ankur Mahtani

Deep Hazardous Events Detection in Top-Down Fish-Eye Images for Railway Applications

Olivier LAURENDIN[1], Anthony FLEURY[2], Sébastien AMBELLOUIS[3],

Sanaa CHAFIK[1], Ankur MAHTANI[1]

[1]FCS RAILENIUM, F-59300 Famars, France
[2]IMT Nord Europe, CERI SN, F-59500 Douai, France
[3]Gustave Eiffel University, Villeneuve d'Ascq, France
Corresponding Author: Olivier Laurendin (olivier.laurendin@railenium.eu)

Abstract

A current trend in railway application research is the development of an autonomous train for regional lines**[1]**. This line of research aims at reducing human input needed to operate a train to optimize train traffic. This in turn could lead to significant improvement in terms of train flow and energy consumption of the railway infrastructure. However, an autonomous train prototype must provide safety guarantees to be put on the market. It must identify safety issues that are currently under a conductor or an on-platform personnel's responsibility. A common source of personal injuries in the railway context are pedestrians stuck in train automatic doors and dragged when the train departs**[2]**. This paper aims at introducing a deep learning solution to identify such safety concerns in due time in addition to current doors obstacle detection systems. We more specifically study the use of an anomaly detection algorithm for this task. These are commonly used in video surveillance systems but their use cases are sensibly different from the vicinity of train doors. A previous work**[1]** introduces a new anomaly detection dataset called FRailTRI20_DOD depicting a set of hazardous events in the vicinity of train doors. This paper proposes a set of modifications to a deep learning-based anomaly detection algorithm of the literature to adapt it to this dataset. Additionally, the proposed modifications are the first work to provide good practices to deal with this dataset specificities.

Keywords: Anomaly Detection ; Deep Learning ; Fish-Eye Cameras ; Automatic Train Doors

1. Introduction

The identification of hazardous events in the wild is a complex task by nature. It is common knowledge that the quality of a deep learning based algorithm is highly dependent on the quantity and representativeness of available data. Several fields of image-related deep-learning algorithms such as object detection algorithms knew an exponential increase in usage and outcomes in the past decade due to the availability of high-scale public datasets**[3]**. Comparatively, hazardous events tend to be very rare instances and are unpredictable by nature. Trying to predict every possible hazardous event in the vicinity of train doors is an arduous task and the time and human resources needed to annotate them is impractical. This prevents the training of a classification algorithm to directly identify hazardous events. Semi-supervised anomaly detection algorithms are an alternative approach which define hazardous instances as outliers from a set of normal instances**[4]**. Such algorithms are specialized in reconstructing a set of frames devoid of anomalies. It is done by training these algorithms to minimize an abnormality function, namely the difference between their input and output frames. This abnormality function is then used during inference to identify anomalous frames: the greater its reconstruction error is, the more likely it is to be abnormal. Hence, this approach only requires a binary label, normal vs anomalous frames, for the testing set.

Several anomaly detection datasets depicting pedestrians are available, such as UCSD**[5]** and UMN**[6]**. Other datasets depict pedestrians in the context of train transportation. The PAMELA-UANDES Dataset**[7]** features pedestrians boarding and alighting a reproduction of an underground carriage taken from a camera placed on the station facing downwards. The BOSS dataset**[8][9]** depicts pedestrians in a moving train acting a set of hazardous scenarios such as people fainting or people fighting taken from a set of cameras placed on the

ceiling of the train. Finally, the FRailTRI20_DOD dataset**[1]**, mainly features passenger exchanges in a train-station interface. It consists of pedestrians opening and closing a reproduction of train doors and crossing the doorway. It also features a set of hazardous events in the vicinity of train doors, see **figure 2**. Namely, it includes doors interrupted while closing, signs of doors mechanical mishaps, pedestrians or pieces of luggage stuck in the doorway and passengers falling. Its frames are taken from a fish-eye camera placed on the ceiling of the train in front of the doors. This last dataset was specifically designed for our application and we will focus on it in the following.

We selected an anomaly detection algorithm from the literature based on the analysis of some key differences between the FRailTRI20_DOD dataset to other datasets. Most anomalies present in other datasets tend to be of a very punctual nature, either as the appearance of a miscellaneous instance or an instance motion pattern sudden change, see **figure 1**. Anomalies depicted in the FRailTRI20_DOD dataset are of a comparatively more complex nature, see **figure 2**. The case of pedestrians stuck in the doorway for instance are the result of an abnormal door-pedestrian interaction. By training on FRailTRI20_DOD a network originally designed for other datasets and analysing its capabilities and shortcomings, we aim to see how well the hazardous events present in this dataset can be summarized to this set of punctual anomalies.

We therefore focused our interest on algorithms whose reconstruction error is a combination of an appearance error and a motion error. In practice, it is often done with the use of auto-encoders or GAN networks which use an input image in addition to its motion in the form of its optical flow, see **figure 3**. While some of such networks such as the works by Ionescu et al.**[10]** and Georgescu et al.**[11]** focus on an object-centric reconstruction error, some other works such as Ravanbakhsh et al.**[12]** and Nguyen et al.**[13]** use a frame-level reconstruction error. We select the network of Nguyen et al. since we want the network to be able to grasp the information of several instances at the same time.

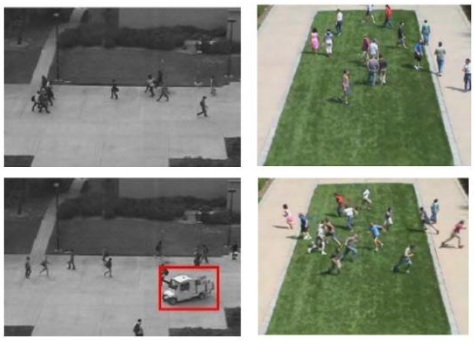

Figure 1: Normal frames and abnormal frames depicted in the UCSD (left) and UMN (right) datasets

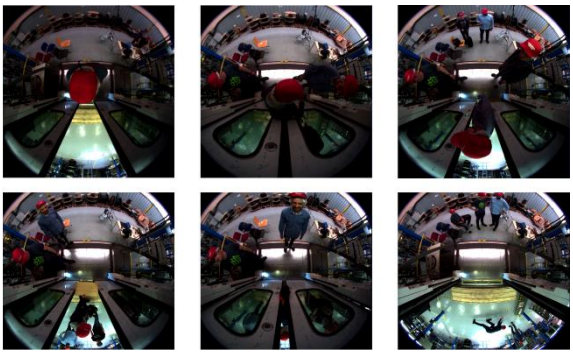

Figure 2: Abnormal frames depicted in the FRailTRI20_DOD dataset.

2. Proposed Method

2.1. Original Network

The functioning of the original network developed by Nguyen et al.**[13]** can be summarized as follows. To predict an anomaly score for a given input image, it is firstly provided to a U-Net auto-encoder called the generator. This generator is composed of an encoder followed by two decoders. The encoder essentially compresses the original image to a lower dimensional feature map which is decompressed by the two decoders. The first decoder is trained to reconstruct the original image while the second infers its optical flow. The reconstruction error of the network is a combination of an appearance reconstruction error, between the input image and the reconstructed image, and a motion error, between the predicted optical flow and the ground truth optical flow provided by Flownet2**[14]**. This network essentially infers instances' displacement

from their appearance. This is particularly well suited to identify anomalous displacement of pedestrians since it is reasonable to assume that a pedestrian walk can be inferred from their stance. To help the second decoder infer the optical flow from the input image, an additional discriminator is added to the network during training. This discriminator will be trained to distinguish between the predicted optical flow of the generator and the ground truth optical flow using the input image as reference. The discriminator thus provides some insights, in the form of an adversarial loss, to the generator of what characteristics distinguishes the predicted optical flow from its ground truth. The generator in turn uses this information to better fool the discriminator. The loss function used to train the generator is therefore the weighted sum of three losses: an appearance loss, an optical flow loss and an adversarial loss. The optical flow loss is the only one modified in the following experiments, other losses in addition to the network overall structure, optimizer and initialization remain untouched from their depiction in the original paper**[13]**.

2. 2. Proposed Modifications

2. 2. 1. Optical flow loss functions

The use of fish-eye cameras in the FRailTRI20_DOD dataset has several impacts on an instance's appearance in the image plane. Firstly, the average size and motion of an instance in the image plane is greater when the instance is in the centre of the image (right under the camera) than on its extremities. We therefore experiment with a set of replacement optical flow loss functions to help the network focus on the moving parts of the image. The first set of approaches consider heuristics to mask off the static parts of the image. It is done by averaging the L1 optical flow error over a restricted portion of the image plane, see **eq. 2**. These approaches include **Radius Masks**, which reduce the area of interest to the pixels at a distance to the centre of the image lower than a given radius (see **Figure 4**) to ignore the heavily distorted extremities of the image. And **Norm Masks** which reduce the area of interest to the pixels where optical flow norm is greater than a given threshold. The optical flow map selected can either be the ground truth optical flow, called **Simple Norm Mask**, or the sum of the ground truth and predicted optical flow map, called **Symmetrical Norm Mask**. The second set of approaches is to replace the L1 norm altogether by the **Ruzicka loss**, see **eq. 3** as introduced in **[1][15]** also known as quantitative Jaccard loss. For a given ground truth and predicted masks, this loss is minimum when these masks match perfectly and does not compute pixels whose labels are simultaneously close to zero. However, since this loss in only defined for positive valued masks, it can only be applied on the norm of the optical flow. The proposed Ruzicka loss for optical flow is provided in **eq. 4**.

1) $L1_{mask}(x,y) = \sum_{mask} abs(x-y) \otimes mask$

2) $L1_{flow} = L1_{mask}(OF_{gt}, OF_{pred})$

3) $Ruzicka(x,y) = 1 - \frac{\sum \min(x,y)}{\sum \max(x,y)}$

4) $Ruzicka_{flow} = Ruzicka(|OF_{gt}|, |OF_{pred}|) + L1_{|OF_{gt}|>\varepsilon}\left(angle(OF_{gt}), angle(OF_{pred})\right)$

$OF: Optical\ Flow\ ; \otimes: hadamard\ product\ ; |.|: vector\ norm\ ; \varepsilon: threshold$

5) $PSNR(x,y) = 10\log_{10}\left(\frac{d^2}{(x-y)^2}\right)$

$d: input\ dynamics$

6) $SSIM(x,y) = \frac{2\mu_x\mu_y}{\mu_x^2+\mu_y^2}\frac{2\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2}\frac{cov_{xy}}{\sigma_x\sigma_y}$

$\mu: mean\ ;\ \sigma: variance\ ;\ cov: covariance$

2. 2. 2. Position-dependant instances appearance

Another consequence of the use of a top-down fisheye camera is that the appearance of a given instance is no longer translation-invariant. As can be seen in **Figure 3**, the appearance of an instance changes from an upright position to an upside-down position when translated along the vertical axis on the image plane. This property goes against the basic assumptions for the usage of convolutional neural networks which are translation-invariant by design. In other words, we want the network to be able to distinguish the cases where a

pedestrian is in an upright position in the top half of the image from a pedestrian in an upright position in the bottom half of the image. Since the pedestrian in the first case is standing up in the 3D space while they would be standing on their head in the second case. The issue of breaking the translation-invariance property of convolutional neural networks was tackled by the paper **[16]** with the use of a **CoordConv layer**. It consists of adding the pixels positions as additional channels to the input data. Convolutional kernels thus process the data contained in the feature map depending on this additional information. We experiment with two kinds of CoordConv layers as the first layer of our network, either using **euclidian** or **polar** coordinates.

2. 2. 3. Additional modifications

We also experimented with data augmentations to artificially increase the diversity of our data. We exploit the symmetries of fish-eye images by randomly flipping images and optical flows horizontally or vertically. Optical flow direction is also reversed accordingly, see **Figure 3**.

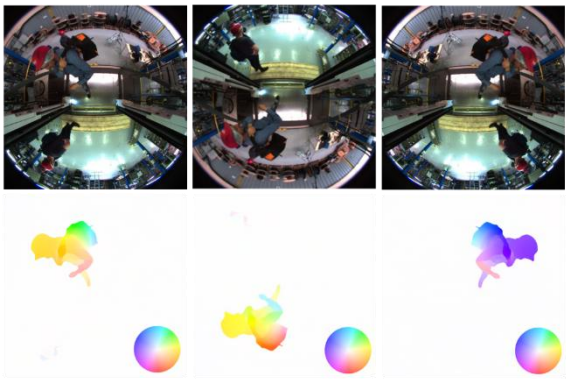

Figure 3: Input images and associated optical flow. The hue indicates its direction while the saturation its norm. 2nd and 3rd columns are flipped images and optical flow along the horizontal and vertical axes.

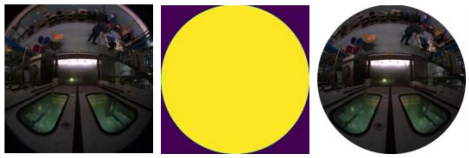

Figure 4: Input image, RadiusMask and cropped image.

2. 3. Experimental Results

Each network was trained for 200 epochs with a batch size of 8 and using the Adam optimizer with initial learning rates of 2e−4 for the generator and 2e−5 for the discriminator. The results are provided in **Table 1** in terms of area under the receiver operating characteristic curve (AUC-ROC) and in terms of average Precision Recall (aPR) of the Peak Signal to Noise Ratio (PSNR, **eq.5.**) and Structural SIMilarity (SSIM, **eq.6.**) of the appearance and motion errors. Predicted and ground truth data are first standardized for best results. Results for a random coin flip in terms of AUC-ROC and aPR for this dataset and the original implementation "**Vanilla**" are also provided. Some qualitative results for the Ruzicka Symmetrical Mask are also provided in **Figure 5** and its ROC and precision-recall curves are shown in **Figure 6**.

	AUCROC				**aPR**			
	Appearance		**Motion**		**Appearance**		**Motion**	
	PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Random	**0.50**				**0.33**			
Vanilla	0.659	0.707	0.741	0.667	0.417	0.454	0.544	0.415
Radius Mask	0.566	**0.807**	**0.805**	0.796	0.353	**0.650**	**0.610**	0.536
Norm Mask (Simple)	0.711	**0.828**	0.566	0.764	0.504	**0.679**	0.404	0.505
Norm Mask (Sym)	0.690	0.770	0.720	0.799	0.462	0.549	0.541	0.534
Ruzicka Mask (Simple)	0.635	0.791	0.767	**0.812**	0.414	**0.623**	0.532	0.562
Ruzicka Mask (Sym)	0.637	0.785	0.768	**0.816**	0.426	0.587	0.529	0.560
Data augmentation	0.585	0.748	0.725	0.658	0.388	0.531	0.499	0.398
Coordconv (euclidian)	0.480	**0.817**	0.768	**0.806**	0.312	**0.652**	0.556	0.538
Coordconv (polar)	0.480	**0.810**	**0.800**	**0.805**	0.324	**0.629**	**0.604**	0.537

Table 1: Experimental results of each implementation. In bold are marked the AUCROC over 0.8 and the aPR over 0.6 while red marks the AUCROC under 0.6 and aPR under 0.4.

Several general observations can be made from these results. Firstly, anomaly detection on average better when using SSIM rather than PSNR, particularly in terms of appearance error. Since PSNR is a pixel-level metric whereas SSIM is patch-level, this seems to indicate the presence of local pixel-level noise which harms image reconstruction. Also, the often-higher results of motion anomaly detection compared to appearance anomaly detection shows, in accordance with the results in [1] that the hazardous events of our use-case are more distinguishable in terms of motion than appearance. However, it should be noted that in all cases, the network is often better at predicting the norm of the optical flow than its direction. As can be seen in **Figure 5**, the network can predict the moving parts of the image, namely where pedestrians and doors are, but is less capable of identifying their motion direction. It can however predict the direction of motion of the doors from a single image, which is a remarkable feat.

On a per-case analysis, the use of the Radius Mask brings better performance in terms of motion anomaly detection in exchange for a partial loss of performance in PSNR appearance anomaly detection. The use of a Simple Norm Mask increases performance across the board except in terms of PSNR motion anomaly detection. While the use of the symmetrical variant smooths the results more evenly. This seems to indicate that some noise induced by the use of the simple mask was mitigated by taking into account the predicted optical flow in the optical flow loss function. While the use of a Ruzicka Mask provides a gain of performance regardless of the use of the simple or symmetrical variant. This shows the ability of the ruzicka loss to faithfully focus on the moving parts of the optical flow. Data augmentation does not increase performance and is actually detrimental. Finally, the use of Coordconv as an input layer has a major impact on motion anomalies detection and appearance anomalies in terms of SSIM. The poor results in terms of appearance PSNR for data augmentation and Coordconv could be caused by a lack of convergence. They will be the subject of further investigation in the future.

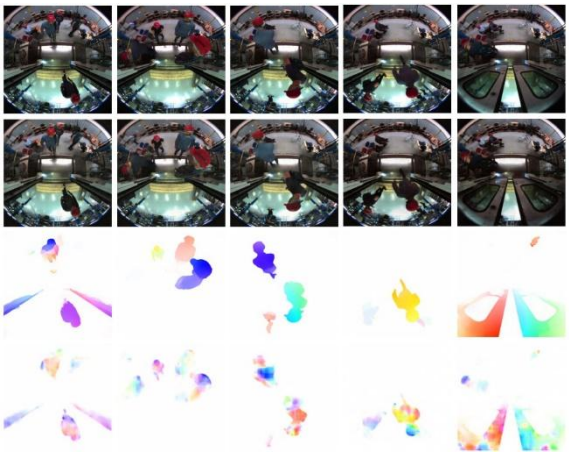

Figure 5: Qualitative results of the Symmetrical Ruzicka Mask network. Each line is respectively the input image, reconstructed image, ground truth optical flow and predicted optical flow.

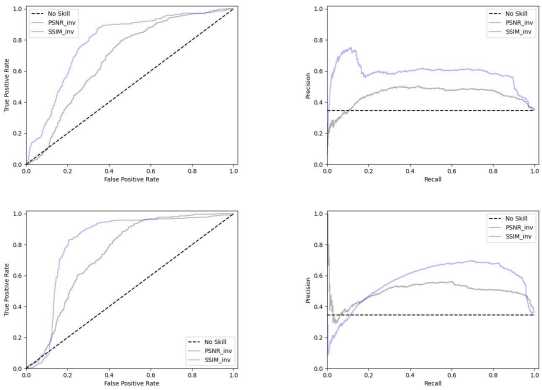

Figure 6: ROC and precision recall curves for the Symmetrical Ruzicka Mask network in appearance error (top line) and motion error (bottom line).

5. Conclusion

After a thorough analysis of the particularities of the FRailTRI20_DOD dataset's data and use-case, a series of modifications to an anomaly detection algorithm of the literature were proposed, several of which had a positive impact on performance. Overall, the use of basic masks on the optical flow maps helps the network focus on the moving parts of the image, sometimes to the detriment of the image reconstruction. This loss of performance in terms of image reconstruction can be mitigated either by incorporating the predicted optical flow in the loss function with a symmetrical mask or by using the Ruzicka loss function. Breaking the translation-invariant property of convolutional kernels also helps the network to identify anomalous events by taking into account its position in the image plane.

Acknowledgment

This research work is funded by the French program ”Investissements d’Avenir” and is part of the French collaborative project TASV (Train Autonome Service Voyageurs), with SNCF, Alstom Crespin, Thales, Bosch, and Spirops.

References

[1] O. Laurendin et al., “Hazardous Events Detection in Automatic Train Doors Vicinity Using Deep Neural Networks,” in *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2021, pp. 1–7.

[2] “Woman dragged under train by bag trapped in door,” *BBC News*, Feb. 29, 2016.

[3] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context,”, 2015.

[4] G. Pang et al., “Deep Learning for Anomaly Detection: A Review,” *ACM Comput. Surv.*, vol. 54, no. 2, pp. 38:1–38:38, 2021.

[5] V. Mahadevan et al., “Anomaly Detection in Crowded Scenes,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1975–1981, 2010.

[6] A. Adam et al., “Robust Real-Time Unusual Event Detection using Multiple Fixed-Location Monitors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 555–560, 2008.

[7] S. A. Velastin, et al., “Detecting, Tracking and Counting People Getting On/Off a Metropolitan Train Using a Standard Video Camera,” *Sensors*, vol. 20, no. 21, p. 6251, 2020.

[8] C. Lamy-Bergot *et al.*, “Transport system architecture for on board wireless secured A/V surveillance and sensing,” in *2009 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST)*, pp. 564–568, 2009.

[9] S. A. Velastin and D. A. Gómez-Lira, “People Detection and Pose Classification Inside a Moving Train Using Computer Vision,” in *Advances in Visual Informatics*, Cham, pp. 319–330, 2017.

[10] R. T. Ionescu et al., “Object-Centric Auto-Encoders and Dummy Anomalies for Abnormal Event Detection in Video,” *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 2019.

[11] M.-I. Georgescu et al., “A Background-Agnostic Framework with Adversarial Training for Abnormal Event Detection in Video,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2021.

[12] M. Ravanbakhsh et al., *“Abnormal event detection in videos using generative adversarial nets,” 2017 IEEE International Conference on Image Processing (ICIP)*, 1577-1581, 2017.

[13] T.-N. Nguyen and J. Meunier, “Anomaly Detection in Video Sequence With Appearance-Motion Correspondence,” *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[14] E. Ilg et al., “FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks,”, pp. 1647–1655, 2017.

[15] M. J. Warrens, “Inequalities Between Similarities for Numerical Data,” *J Classif*, vol. 33, no. 1, pp. 141–148, 2016.

[16] R. Liu et al., “An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution”, *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.