Anomaly Detection, Localization and Classification for Railway Inspection

Riccardo Gasparini[1], Andrea D'Eusanio[1], Guido Borghi[1],
Stefano Pini[1], Giuseppe Scaglione[2], Simone Calderara[1], Eugenio Fedeli[2], Rita Cucchiara[1]
[1]AIRI - Artificial Intelligence Research and Innovation Center, Università di Modena e Reggio Emilia
[2]RFI - Rete Ferroviaria Italiana, Gruppo Ferrovie dello Stato, Firenze
Email: [1]{name.surname, s.pini}@unimore.it
[2]{g.scaglione, e.fedeli}@rfi.it

***Abstract*—The ability to detect, localize and classify objects that are anomalies is a challenging task in the computer vision community. In this paper, we tackle these tasks developing a framework to automatically inspect the railway during the night. Specifically, it is able to predict the presence, the image coordinates and the class of obstacles. To deal with the low-light environment, the framework is based on thermal images and consists of three different modules that address the problem of detecting anomalies, predicting their image coordinates and classifying them. Moreover, due to the absolute lack of publicly-released datasets collected in the railway context for anomaly detection, we introduce a new multi-modal dataset, acquired from a rail drone, used to evaluate the proposed framework. Experimental results confirm the accuracy of the framework and its suitability, in terms of computational load, performance, and inference time, to be implemented on a self-powered inspection system.**

I. Introduction

Anomaly detection is defined as the identification of samples which exhibit significant differences with respect to the regularity. The ability of autonomous systems to detect and recognize unknown objects or events is crucial in many application domains, ranging from defect detection [1], video surveillance [2], medical imaging [3] to reinforcement learning [4]. The large majority of literature works about anomaly detection assume that acquisition devices are in a fixed position and then images and videos have static backgrounds [5]. For instance, this is the case of methods relying on data collected by video surveillance [6] or industrial cameras [7].
Moreover, several works perform anomaly detection relying on a supervised approach [8], [9], that usually requires time-consuming manual annotations, together to the unrealistic assumption that all anomaly patterns are available during the training process. Only recently, some works investigate the anomaly detection task on videos acquired through a dashboard-mounted camera on a moving ego-vehicle for traffic accident detection on roads [5].

In this paper, we propose a framework for the anomaly detection task on videos acquired from a moving camera. Specifically, our framework is able to detect, *i.e.* predict if a frame is regular or anomalous, to localize, *i.e.* predict the image coordinates, and to identify, *i.e.* predict the class object, anomalies in thermal video sequences, as shown in Figure 2. The framework is created for the automatic inspection of railways based on thermal images and the vision acquisition system is implemented on a rail drone, a small and light-weight vehicle, operated by remote control. Thus, a reduced energy consumption and real time performance are required.
The inspection of railways, *i.e.* the activity to check the absence of obstacles placed on the railroad that could damage or derail trains, is a key element to guarantee the safety of transports. Due to the vastness of railways, an automatic inspection conducted during nighttime, when the train circulation is usually suspended, is strongly demanded.
Therefore, we collect a new dataset, namely *Vesuvio*, that contains more than 30k frames acquired during the night through a vision system based on a synchronized stereo, thermal and standard RGB cameras. The dataset contains more than 50 anomalous and regular (non-anomalous) sequences. Anomalies consist in various objects, usually employed in rail yards, and depicted in Figure 1, in the thermal domain.
Summarizing, the main contributions are the following:

- We present a new framework for the automatic inspection of railways during the night. The framework is based on thermal images and consists of three different modules to detect, localize and classify anomalies in video sequences.
- We present a new dataset, *Vesuvio*, specifically created for the anomaly detection task with moving cameras. To the best of our knowledge, this is the first publicly-released dataset acquired from a rail drone during the night.
- The proposed framework achieves good accuracy and real time performance, representing a suitable solution for a self-powered system installed on a rail drone.

II. Related Work

Anomaly Detection Generally, literature works that address the anomaly detection task are divided in two different approaches: reconstruction-based models and probabilistic methods. The former propose to learn a parametric reconstruction of normal data, through traditional sparse-coding algorithms [10], [11], deep autoencoders [2] or generative adversarial networks [6]. A similar approach is the future frame prediction: in [12] anomalies are detected comparing the differences between a predicted future frame and the current acquired frame.

The latter rely on the approximation of a density function of motion features and normal appearance. In this case, optical flow and trajectory analysis exploiting non-parametric [13] and parametric [14] estimators are usually used. Recently, the introduction of deep learning-based representations has grown [3], [15].
Even though methods for anomaly detection with fixed cameras achieve the state-of-the-art accuracy, highly dynamic scenarios, such as the railway inspection with images taken from a drone, pose challenges in reconstruct what is considered as regular [5]. Besides, we observe that these types of work are mainly focused on video surveillance scenarios [16], [17].
Only a minor part of the works on anomaly detection are based on moving cameras. In [5], an unsupervised approach is proposed for traffic accident detection. The vision acquisition system is a dashboard-mounted camera and the key idea of this method is to predict the future locations of traffic participants in order to avoid car crashes. In [18], a dataset of crowd-sourced dash camera images is presented and a supervised method to detect anomalies, in terms of driving offences and motorbike and car collisions on roads, is proposed. Abati *et al*. [19] proposed an anomaly detection method on an automotive dataset [20], but the visual content is purposely discarded, maintaining only eye fixations.

Anomaly Detection on Railways In general, a very limited amount of works exploit visual data in the railway scenario, for similar tasks such as anti-collision prediction [21], [22] and track detection [23], [24], [25]. Only few works tackle the task of anomaly detection applied on this specific context. Unfortunately, we note that datasets are often not publicly available.
Usually, literature works exploit the use of infrared or ultrasonic range sensors placed on trains. In [26], a system based on a range sensor is proposed in order to perform obstacle detection. An infrared emitter is placed in the frontal part of the locomotive and a light turns on when an object is detected within a range distance. In [27] authors proposed a framework based on infrared sensors and GSM and GPS signals addressing the obstacle detection and avoidance and the train tracking. Similar to the previous work, an infrared sensor is used to detect the presence of obstacles in front of the locomotive. In [28] a Lidar and a vision system are implemented, the two sensor outputs are then merged to detect frontal obstacles.
Only recently, in [29] a public dataset for semantic scene understanding for trains and trams, namely *RailSem19*, is introduced. This dataset covers a variety of tasks, such as classification of trains, switches, switch states, platforms, buffer stops, rail traffic signs and rail traffic lights. Unfortunately, obstacles have not been considered. In [30], infrared sensors are not placed on the train, but on the railway sides: the lack of connection between emitters and receivers reveals the presence of obstacles.
In general, we note that solutions based only on a vision system are not present in the literature. In addition, there is a lack of datasets acquired on railways through a vision-based system.

III. *Vesuvio* DATASET

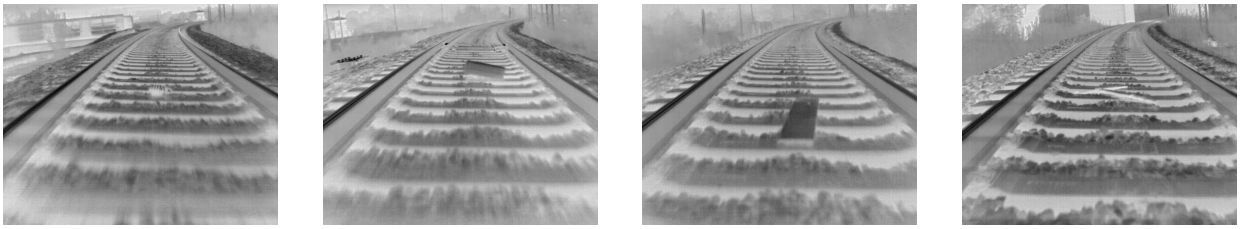
To the best of our knowledge, this is the first dataset collected for the anomaly detection task in a railway scenario during the night. As mentioned above, the dataset is acquired placing a variety of cameras in the front part of a rail drone, so cameras are placed very close to the cobbled road.
Data is collected during the night since the inspection activities are planned to be done when the train circulation is usually suspended.
Considering the aforementioned elements, there are three main aspects that have to be taken into account for the choice of the acquisition devices, directly derived from the automotive context [31]:

- **Night Vision**: acquisition cameras have to deal with the night time of the acquisition process [32]. This issue is tackled with the adoption of external light sources and thermal cameras. It is important to note that there are limitations in the power consumption of light sources since the rail drone is self-powered.
- **Fast acquisition**: the frame rate, expressed in terms of frames per second (fps), and the shutter speed of the cameras must be high enough to avoid motion blur [33] caused by the high speed of the drone (up to 100 km/h).
- **High Resolution**: in order to guarantee that even small-size objects are detected by the vision system, acquisition cameras with a high spatial resolution are needed.

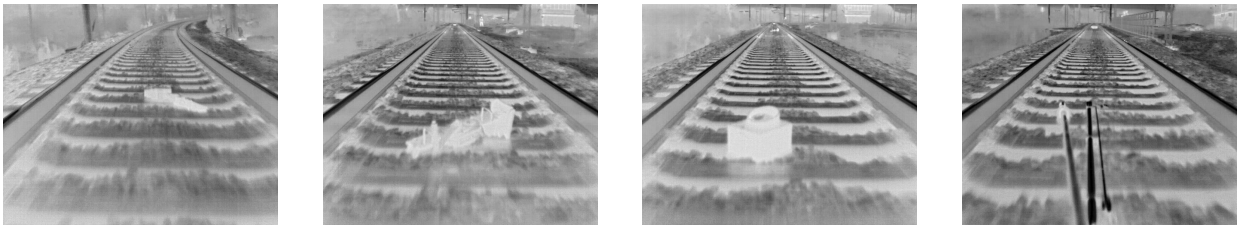
To comply with these requirements, the following cameras are employed:

- **Flir Boson 640**[1]: this is a high-resolution thermal camera which is able to acquire frames with a spatial resolution of 640×480 pixels, up to 60 frames per second. This camera, due to its form factor ($21 \times 21 \times 11$ mm), weight (7.5g) and limited energy consumption (only 500mW) is particularly suitable to be installed on the rail drone. The camera is equipped with a 14mm lens.
- **Zed stereo camera**[2]: this is a stereo camera specifically created for the outdoor setting. It has a resolution of 4416×1242 pixels and it is able to acquire 3D surroundings up to 20 meters of distance, ranging from 15 to 100 frames per second depending on the resolution. It needs a dedicated graphics processing unit to run in real time and an external light source.
- **Basler acA800-510uc**[3]: this industrial RGB camera has a high frame rate (500 fps) that imposes a limited spatial resolution of 800×600 pixels. The camera is equipped with a 75mm zoom lens. Also in this case, an external light source is required.

[1] https://prod.flir.it/products/boson
[2] https://www.stereolabs.com/zed
[3] https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca800-510uc



(a) Electrical Insulator (b) Fuel tank (c) Rail signal (d) Pickaxe



(e) Locking turnout (f) Track lifting jack (g) Traffic light (h) Insulating stick

(i) LPG tank (j) Balise (k) Oiler (l) No anomaly

Fig. 1: Anomaly classes included in the *Vesuvio* dataset. Images of the thermal domain.

The *Vesuvio* dataset consists of more than 10k frames. Every frame is manually annotated with the presence and the location (bounding box) of obstacles. There are 11 classes of anomalies, *i.e.* objects placed on the railway:

- Electrical insulator
- Fuel tank
- Rail signal
- Pickaxe
- Locking turnout
- Track lifting jack
- Traffic light
- Insulating stick
- LPG tank
- Balise
- Oiler

All classes are depicted in Figure 1 in the thermal domain (the domain exploited in the proposed framework). These objects are the common tools that are in construction sites along the railways.

IV. Proposed Framework

We propose a framework consisting of 3 different modules, each specialized in a different task: anomaly detection, anomaly localization and anomaly classification, as shown in Figure 2. Frames acquired by the vision system, placed in the frontal part of the rail drone, are the input data of the architecture. In particular, in this paper we employ the frames collected through the thermal camera. We detail each module, in terms of task, architecture and training procedure, in the following paragraphs.

A. Anomaly Detection Module

Taking inspiration from [34], the first module of the framework performs the anomaly detection task. In our case, the goal is to predict if an acquired frame contains or not anomalies, *i.e.* obstacles on the rails.
The input of the module is a single thermal frame. Since cameras are placed in a fixed position on the drone, their height from soil is always the same. Therefore, input images are cropped to discard meaningless areas of the acquired frame (*i.e.* area outside the railways, see Fig. 2). The output of the module is represented by a binary frame label.
The model consists of 2 different networks: their size is limited to balance detection accuracy, inference speed and energy consumption. The first network is a deep encoder-decoder architecture [35], here referred merely as autoencoder, that receives as input only regular frames (during the training) and outputs the reconstructed ones. Then, such reconstruction should represent a clean image devoid of any anomaly. This reconstruction is, then, compared with the input through an absolute and a gradient difference, *i.e.* a difference computed only on the gradients, of the two images. The 2 resulting images are then stacked as a dual-channel image and used as input for the second network of the module, that predicts the presence or the absence of anomalies into the frame. The use of difference images as input leads this second network to use both the information in terms of difference in textures and

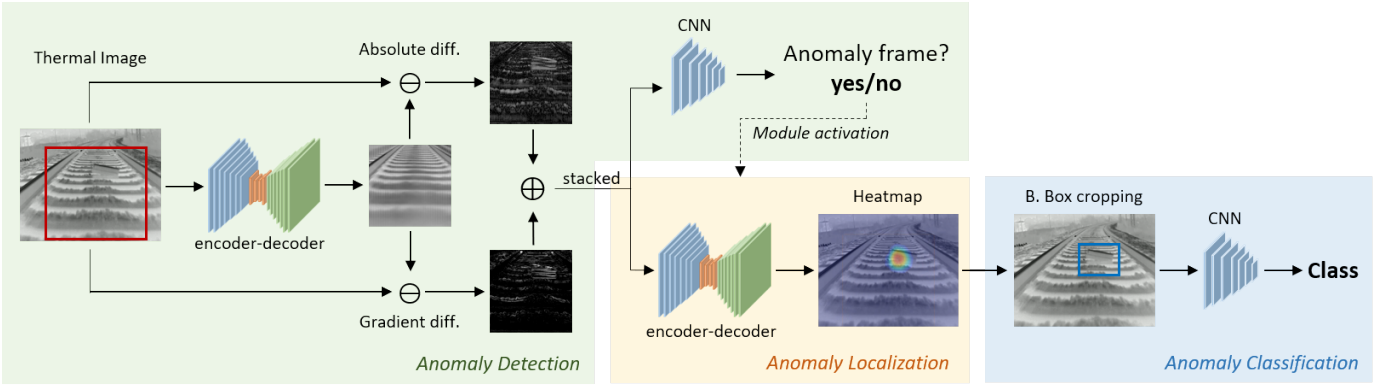


Fig. 2: Overall view of the proposed system. In green, the *Anomaly Detection* module: given a thermal frame as input, the module predict if the frame contains or not an anomaly. If yes, the module of *Anomaly Localization* (orange) is activated and predicts the localization of the anomaly in the given input frame. Finally, the *Anomaly Classification* module (blue) identifies the class of the anomaly. Details about the deep architectures are reported in Section IV.

patches (absolute difference) and the difference in contours and lines (gradient distance).
These steps are reported in the green part of figure 2.

Model. As mentioned above, this module consists in 2 different architectures and accepts input images with a resolution of 192×192 pixels. The first network is an encoder-decoder model, in which the encoder part consists in 9 convolutional layers with stride $s = 2$, only the first and the last two layers have stride $s = 1$. The decoder part is symmetrical: it is composed of 9 transpose convolutional layer to up-sample the feature maps, first two and the last have stride $s = 1$, the remaining $s = 2$. All layers of the encoder and decoder parts have kernel size $k = 3$. Except for the first layer, the size of the feature map is doubled (and then halved) at each layer, starting from 16, arriving to 1024 in the bottleneck, then back to 16 again at the end of the decoder. The final output is still an image with size 192×192 pixels. For all layers, *Leaky ReLu* [36] with slope $s = 10^{-2}$ is used as activation function. This deep architecture has $\simeq$ 22M parameters.
The second network of the module is a CNN that has the same architecture of the previous encoder, but the number of filters is halved (from 8 to 256) and the last convolutional layer is removed. After this layer, feature maps are flatten and 2 linear layers are added in order to output the binary classification (anomaly or not-anomaly). The first linear layer has 48 units, while the second one has 2 units. There is a dropout regularization layer with drop probability $p = 0.3$ in the middle. This network contains $\simeq$ 700k parameters.

Training. The encoder-decoder architecture is trained with an unsupervised approach. During the training, the autoencoder receives only frames without anomalies.
Since the encoder-decoder architecture aims to reconstruct the input frame, we adopt the commonly used *Mean Squared Error* (L_{MSE}) as loss function:

$$L_{\text{MSE}} = \frac{1}{MN} \sum_{m}^{M} \sum_{n}^{N} \| I_i(m,n) - I_r(m,n) \|_2^2 \qquad (1)$$

In addition, we use a Gradient Loss (L_{GL}) defined as:

$$G_x = I * \begin{vmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{vmatrix}, \quad G_y = I * \begin{vmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{vmatrix} \qquad (2)$$

$$G = \sqrt{G_x^2 + G_y^2} \qquad (3)$$

$$L_{\text{GL}} = \frac{1}{MN} \sum_{m}^{M} \sum_{n}^{N} \| G_{I_i}(m,n) - G_{I_r}(m,n) \|_2^2 \qquad (4)$$

In Equations 1 and 4, I of size $M \times N$ is the input (I_i) of the reconstructed (I_r) image while G_{I_i} and G_{I_r} are the gradients computed respectively on the input and the reconstructed images.
Equations 2 and 3, firstly described in [37], allow the computation of the gradient along both vertical and horizontal dimensions of an image. Minimizing this loss is equivalent to improve the definition of lines and contours in the reconstructed frame.
Taking inspiration from [38], the final loss L used in the training procedure is:

$$L = \alpha \cdot L_{\text{MSE}} + \beta \cdot L_{\text{GL}} \qquad (5)$$

In our experiments, we set $\alpha = \beta = 1$, while the learning rate is set to $lr = 10^{-3}$.
For the second architecture, the *Binary Cross Entropy* loss function is exploited with a learning rate of $lr = 10^{-2}$. The Adam [39] optimizer is used for the training of both the architectures.

B. Anomaly Localization Module

The goal of this module is to localize the detected anomaly in frame coordinates. This module is activated only when the first one classifies the frame as anomaly. The input is the dual-channel image computed by the previous module as the stack of the absolute and the gradient difference. The output is a probability map, here referred also as *heatmap*, *i.e.* a map

TABLE I: Experimental results for the proposed framework. The framework is tested on two dataset splits: "80-20" and "cross-class". Details are reported in Section V.

Modules	80-20 split				Cross-class split			
	Accuracy	**Precision**	**Recall**	**F1-score**	**Accuracy**	**Precision**	**Recall**	**F1-score**
Anomaly Detection	0.966	0.989	0.957	0.973	0.848	0.903	0.776	0.835
Anomaly Localization	0.903	0.741	0.989	0.847	0.785	0.521	0.997	0.684
Anomaly Classification	0.970	0.795	0.794	0.785	-	-	-	-

in which the location of the anomaly is expressed with a bi-variate Gaussian function whose peak is centered on the anomaly. The probability map is then used to extract a crop of the detected anomaly which will be classified by the last module of the framework. This approach allows to overcome the traditional sliding-window approach, producing benefits in speed performance and detection accuracy. The module is represented in the orange area of Figure 2.

Model. The network is based on an encoder-decoder architecture which is similar to the one exploited in the anomaly detection module, but differs in some details. In particular, the number of filters are 8 for the input and the output layer, the bottleneck have size of 512 and max-pooling layers (with kernel size $k = 3$ and stride $s = 2$) are used instead of convolutional layers with stride $s = 2$. The generated heatmap has the size of the input image, *i.e.* 192×192 pixels.

The resulting autoencoder is lighter, in terms of number of parameters, than the one used in the first module. Indeed, the aim of this architecture is not the full reconstruction of the input image, but the generation of a probability map with a Gaussian function centered on the anomalous object. The network consists of $\simeq 5.5$M parameters.

Training. In this case, we employ a supervised training: the ground-truth probability maps are generated as maps of shape 192×192 pixels on which we apply a bi-variate Gaussian function, centered on the anomalies which have been manually annotated with bounding boxes in the *Vesuvio* dataset. In our experiments, we set $\sigma = 0.25 \cdot A$, where A is the area in pixels of the rectangular bounding box. The loss used is the MSE, as detailed in Equation 1, but computed between the predicted and the ground truth heatmap, with a learning rate $lr = 10^{-4}$ and the Adam [39] optimizer.

C. Anomaly Classification Module

The last module of the framework aims to classify a detected and localized anomaly. Therefore, the deep architecture of this module acts as a multi-class classifier. The input is the heatmap computed by the previous module and the initial thermal image. Through the heatmap, the thermal image is cropped at the anomaly location (see Section V-A for further details) and only this portion of the image is classified by the architecture. The network outputs a probability score on the list of classes described in Section III and the output of the module is the class with the higher probability. The module is represented in the blue area of Figure 2.

TABLE II: Results of the anomaly classification module.

Class	Accuracy	Precision	Recall	F1-score
Electrical insulator	0.94	0.54	0.44	0.48
Fuel tank	0.96	0.65	0.76	0.70
Rail signal	0.99	1.0	0.77	0.87
Pickaxe	0.97	0.58	0.92	0.71
Locking turnout	0.99	0.88	0.78	0.82
Track lifting jack	0.98	0.81	0.94	0.87
Traffic light	0.99	0.88	0.95	0.91
Insulating stick	0.98	0.97	0.90	0.94
LPG tank	0.97	0.92	0.79	0.85
Balise	0.93	0.73	0.53	0.62
Olier	0.97	0.79	0.96	0.87

Model. The model is based on a Convolution Neural Network that is equivalent of the encoder block of the autoencoder architecture employed in the second module. The last convolutional layer is replaced with 2 linear layers with size 16 and 12 (equal to the number of classes), respectively. The input size is fixed to 64×64: smaller images are zero-padded while bigger images are appropriately resized (the bigger side is resized to 64 while the other side is zero-padded). This module has only 600k parameters.

Training. For the training procedure we exploit the class annotations in terms of bounding boxes provided in the *Vesuvio* dataset. Parameters are optimized by Adam [39] with an initial learning rate of 10^{-4}, while the *Categorical Cross Entropy* is exploited as loss function.

V. Experimental Evaluation

In this Section, we conduct the experimental evaluation of the proposed framework. Firstly, we report the metrics exploited to assess the quality of our method. Then, we collect result for each single module, in order to understand the contribution of each module. Finally, we report the results of the whole pipeline, in terms of accuracy and inference time.

We test the proposed framework on two different settings of the dataset.

In the first one, we group all the anomaly frames, *i.e.* frames that contain an object. Then, for each class, we sample about 80% of frames for the training step and the remaining 20% for the testing phase. Finally, we randomly sample from the original dataset an equal number of regular frames both for the training and testing subsets. We refer to this common dataset setting as "80-20".

The second setting is a more-challenging *cross-class* modality: we select 3 classes which differ for their appearance (pickaxe,

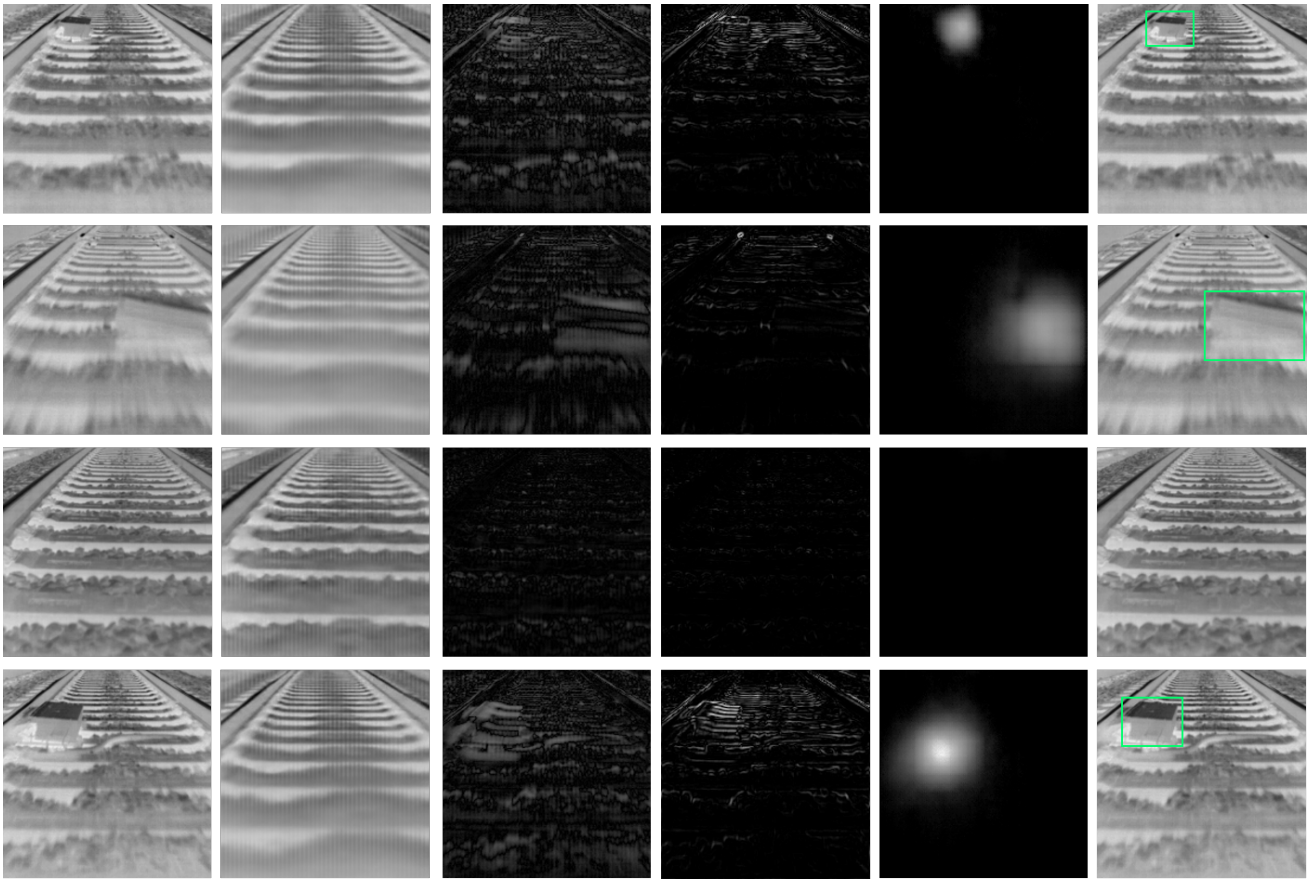


Fig. 3: Output of the proposed framework. From the left are reported: the thermal input frame, the reconstructed frame, the absolute and gradient difference images, the heatmap and finally the bounding box (here represented as a green rectangle) processed by the anomaly classification module. In the third row a regular frame, *i.e.* a frame without anomalies, is reported.

fuel tank and oiler) for the testing set, and the remaining 8 classes for the training set. Similar to the previous case, we then sample an equal number of regular frames from the dataset for both training and testing parts.

A. Metrics

For all the experiments, we use the common accuracy, precision, recall and F1 metrics. The anomaly detection is a binary classification task, since a frame is predict as regular or as anomaly, while the anomaly classification is a multi-class task, since there are 12 classes in *Vesuvio* dataset (11 anomalies and 1 class for frames without any object).
For the anomaly localization results, we consider an object as correctly located if:

$$IoU(A, B) > \tau \tag{6}$$

where

$$IoU(A, B) = \frac{\textit{Area of Intersection}}{\textit{Area of Union}} = \frac{|A \cap B|}{|A \cup B| - |A \cap B|} \tag{7}$$

in which A and B are ground truth and predicted anomaly bounding boxes, respectively, while the threshold τ is set to 0.3 as used in [40]. Bounding box A is provided in the dataset annotations, while B is computed by finding the peak of the Gaussian function in the heatmap and then calculating the bounding box size as $3 \cdot \sigma$. Moreover, we compute the average distance d between the centers of the bounding boxes A and B, or rather:

$$d = \frac{1}{N} \sum \|c_A - c_B\|_2 \tag{8}$$

where $c_{A,B}$ are the centers of the bounding boxes A and B while N is the number of test frames.

B. Results

Results for the anomaly detection, localization and classification tasks are reported in Table I. In general, experimental results confirm that the proposed framework is able to handle objects never seen during the training phase.

For the anomaly detection module, we observe that the framework is able to achieve a good accuracy, both in terms of precision and recall. Thermal data are probably a good choice and the classification of the difference images (absolute and gradient differences between input and reconstructed frames) represents a suitable approach in order to detect anomalies on the railways. For the anomaly localization module, in addition to the results in Table I, we report an average distance of

$d = 7.3$ pixels for the "80-20" split and an average distance of $d = 16.6$ pixels in the cross-class modality.

For the anomaly classification task we maintain only the "80-20" setting, since the classifier is trained and tested on all classes. General results are reported in Table I, while results focused on each class are reported in Table II. We observe that the precision and recall scores are negatively influenced by the size of the anomaly: for instance, the electrical insulator is the smallest anomaly in the dataset. Also objects that are similar, from a visual point of view, to elements typically on the railway (such as sleepers) present low scores: this is the case of the pickaxe and balises. Furthemore, we note that the adoption of a shallow network for the classification task does not significantly compromise the final performance.

Finally, we test each module for the evaluation of the inference time. The framework achieves 190 fps for the anomaly detection module, 130 fps for the anomaly localization and more than 1000 fps for the object classification task. Overall, the system is able to run at about 100 fps considering the whole pipeline, *i.e*. all modules running at the same time. This is due to the adoption of architectures that are balanced between the number of parameters, *i.e*. the computational load, and the final accuracy. Tests have been carried out on a PC equipped with an Intel *i7-4790* CPU (3.60 GHz) and a *NVidia P4000*. The deep architectures of the framework have been implemented in *Pytorch*.

VI. Conclusion

In this paper, we proposed a new challenging dataset, namely *Vesuvio*, and a multi-stage framework to perform detection, localization and classification of anomalies on railways, in order to detect obstacles that could affect the safety of the train transport. Three different acquisition devices – a thermal, a stereo and a RGB camera – have been placed on a rail drone and images have been acquired during nighttime. Experimental results confirm the effectiveness of the proposed method Real-time performance is achieved thanks to the use of shallow deep architectures, balancing a low computational load and high accuracy. Future work will regard the use of the stereo data and the intensity images, available in the dataset, in conjunction with thermal data, to improve the overall accuracy and the reliability of the system. Furthermore, as reported in [41], more accurate predictions may be obtained exploiting the time information embedded in the video acquired.

Acknowledgments

We thank Ivan Mazzoni (RFI), Marco Plano (RFI) e Mattia Bevere (RFI) for the technical support.

References

[1] A. Kumar, "Computer-vision-based fabric defect detection: A survey," *IEEE transactions on industrial electronics*, vol. 55, no. 1, pp. 348–363, 2008. 1

[2] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 733–742. 1

[3] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*. Springer, 2017, pp. 146–157. 1, 2

[4] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 16–17. 1

[5] Y. Yao, M. Xu, Y. Wang, D. J. Crandall, and E. M. Atkins, "Unsupervised traffic accident detection in first-person videos," *arXiv preprint arXiv:1903.00618*, 2019. 1, 2

[6] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3379–3388. 1

[7] D. P. Filev, R. B. Chinnam, F. Tseng, and P. Baruah, "An industrial strength novelty detection framework for autonomous equipment monitoring and diagnostics," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 4, pp. 767–779, 2010. 1

[8] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld, "Toward supervised anomaly detection," *Journal of Artificial Intelligence Research*, vol. 46, pp. 235–262, 2013. 1

[9] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 622–637. 1

[10] B. Zhao, L. Fei-Fei, and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *CVPR 2011*. IEEE, 2011, pp. 3313–3320. 1

[11] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *CVPR 2011*. IEEE, 2011, pp. 3449–3456. 1

[12] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection–a new baseline," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6536–6545. 1

[13] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "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. 2

[14] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8. 2

[15] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe, "Training adversarial discriminators for cross-channel abnormal event detection in crowds," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1896–1904. 2

[16] Y. S. Chong and Y. H. Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," in *International Symposium on Neural Networks*. Springer, 2017, pp. 189–196. 2

[17] J. R. Medel and A. Savakis, "Anomaly detection in video using predictive convolutional long short-term memory networks," *arXiv preprint arXiv:1612.00390*, 2016. 2

[18] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 136–153. 2

[19] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent space autoregression for novelty detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 481–490. 2

[20] A. Palazzi, D. Abati, F. Solera, R. Cucchiara *et al.*, "Predicting the driver's focus of attention: the dr (eye) ve project," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1720–1733, 2018. 2

[21] F. Maire, "Vision based anti-collision system for rail track maintenance vehicles," in *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2007, pp. 170–175. 2

[22] F. Maire and A. Bigdeli, "Obstacle-free range determination for rail track maintenance vehicles," in *2010 11th International Conference on Control Automation Robotics & Vision*. IEEE, 2010, pp. 2172–2178. 2

[23] J. C. Espino and B. Stanciulescu, "Rail extraction technique using gradient information and a priori shape model," in *2012 15th International*

IEEE Conference on Intelligent Transportation Systems. IEEE, 2012, pp. 1132–1136. 2

[24] M. Gschwandtner, W. Pree, and A. Uhl, "Track detection for autonomous trains," in *International Symposium on Visual Computing*. Springer, 2010, pp. 19–28. 2

[25] M. H. Zwemer, D. W. van de Wouw, E. G. Jaspers, S. Zinger *et al.*, "A vision-based approach for tramway rail extraction," in *Video Surveillance and Transportation Imaging Applications 2015*, vol. 9407. International Society for Optics and Photonics, 2015, p. 94070R. 2

[26] R. Passarella, B. Tutuko, and A. P. Prasetyo, "Design concept of train obstacle detection system in indonesia," *IJRRAS*, vol. 9, no. 3, pp. 453–460, 2011. 2

[27] N. S. Punekar and A. A. Raut, "Improving railway safety with obstacle detection and tracking system using gps-gsm model," *International Journal of Scientific & Engineering Research*, vol. 4, no. 8, pp. 282–288, 2013. 2

[28] S. Mockel, F. Scherer, and P. F. Schuster, "Multi-sensor obstacle detection on railway tracks," in *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683)*. IEEE, 2003, pp. 42–46. 2

[29] O. Zendel, M. Murschitz, M. Zeilinger, D. Steininger, S. Abbasi, and C. Beleznai, "Railsem19: A dataset for semantic rail scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0. 2

[30] J. J. García, C. Losada, F. Espinosa, J. Ureña, Á. Hernández, M. Mazo, C. de Marziani, A. Jiménez, E. Bueno, and F. Álvarez, "Dedicated smart ir barrier for obstacle detection in railways," in *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.* IEEE, 2005, pp. 6–pp. 2

[31] E. Frigieri, G. Borghi, R. Vezzani, and R. Cucchiara, "Fast and accurate facial landmark localization in depth images for in-car applications," in *International Conference on Image Analysis and Processing*. Springer, 2017, pp. 539–549. 2

[32] M. Venturelli, G. Borghi, R. Vezzani, and R. Cucchiara, "Deep head pose estimation from depth data for in-car automotive applications," in *International Workshop on Understanding Human Activities through 3D Sensors*. Springer, 2016, pp. 74–85. 2

[33] A. D'Eusanio, A. Simoni, S. Pini, G. Borghi, R. Vezzani, and R. Cucchiara, "Multimodal hand gesture classification for the human–car interaction," in *Informatics*, vol. 7, no. 3. Multidisciplinary Digital Publishing Institute, 2020, p. 31. 2

[34] R. Gasparini, S. Pini, G. Borghi, G. Scaglione, S. Calderara, E. Fedeli, and R. Cucchiara, "Anomaly detection for vision-based railway inspection," in *European Dependable Computing Conference*. Springer, 2020, pp. 56–67. 3

[35] G. Borghi, M. Fabbri, R. Vezzani, R. Cucchiara *et al.*, "Face-from-depth for head pose estimation on depth images," *IEEE transactions on pattern analysis and machine intelligence*, 2018. 3

[36] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015. 4

[37] J. M. Prewitt, "Object enhancement and extraction," *Picture processing and Psychopictorics*, vol. 10, no. 1, pp. 15–19, 1970. 4

[38] G. Borghi, S. Pini, R. Vezzani, and R. Cucchiara, "Driver face verification with depth maps," *Sensors*, vol. 19, no. 15, p. 3361, 2019. 4

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 4, 5

[40] D. Ballotta, G. Borghi, R. Vezzani, and R. Cucchiara, "Head detection with depth images in the wild," in *13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS, 2017. 6

[41] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE transactions on vehicular technology*, vol. 69, no. 1, pp. 41–54, 2019. 7