ISSN 2593-7568

RESEARCH ARTICLE

Application of Belief Theories for Railway Track Defect Detection

Alain Rivero[1*], Sasa Radosavljevic[1], Philippe Vanheeghe[2]

[1]SNCF-Reseau, Direction Generale Industrielle et Ingenierie, Department IP3M DM MATRICE, Paris, France
[2]Univ. Lille, CNRS, Centrale Lille, UMR 9189 – CRIStAL, Lille, France

Abstract

Faced with increasing traffic, railway infrastructures are encountering growing demands, particularly in high-traffic areas. In this context, rail and sleepers emerge as the components most susceptible to failure. To assist infrastructure managers (IM) in optimizing network maintenance, we have explored a novel method for detecting critical defects on the track. The objective is to develop a process for real-time analysis of railway infrastructure that is both frugal and efficient and can be installed on board commercial trains. This new infrastructure monitoring system integrates deep learning networks with a data fusion model based on belief theory. By modeling the decision-making process of a human operator, this processing chain has achieved detection rates exceeding 90% for the five primary defects: defective fasteners, broken fishplates and rails, surface defects, and missing nuts.

Key Words: *Belief theory; Yolo network; UberNet architecture; Data fusion; Infrastructure monitoring*

1. Introduction

Pattern recognition problems quickly encounter the phenomenon of combinatorial explosion. While many strategies have been successfully employed to extend the capabilities of these methods, they often revert to the original problem as the domain expertise broadens. In the railway sector, the complexities associated with detecting track defects pose challenges for comprehensive understanding, prompting us to adopt a localized approach. This involves distributing the problem across a set of deep networks (CNN). Then coordinating their outcomes through a fusion model allows us to achieve a decision closely aligned with reality. Our research demonstrates the superiority of this approach in identifying rail defects. This architecture offers tangible benefits over traditional image processing detection techniques, such as Histograms of Oriented Gradients (HOG), wavelet processing [1-3], and Gabor filters [4], among others.

***Corresponding Author**: *Alain Rivero, Project Manager, SNCF-Reseau, Direction Generale Industrielle et Ingenierie, Department IP3M DM MATRICE, Paris, France; E-mail: alain.rivero@reseau.sncf.fr*

***Received Date**: April 26, 2024, **Accepted Date**: May 22, 2024, **Published Date**: May 27, 2024*

***Citation:** Rivero A, Radosavljevic S, Vanheeghe P. Application of Belief Theories for Railway Track Defect Detection. Int J Auto AI Mach Learn. 2024;4(1):10-35*

While these methods have yielded positive results and often led to the commercialization of several monitoring systems, they suffer from significant drawbacks: generating numerous false positives and necessitating complex and expensive maintenance of the encoded monitoring systems. Conversely, techniques employing CNN and fusion models offer the following advantages:

- They can recognize shapes that classical approaches cannot.
- They are adaptable to environmental variations, accommodating novelty.
- They can analyze extremely degraded signals.
- They enable simplified maintenance focused solely on new learning.

Operators aboard infrastructure monitoring trains identify faults by cross-referencing information within an image. The presence or absence of contextual elements informs their decisions, either validating or invalidating the defect. We modeled this human thought process to develop an efficient defect detection system. This approach led to the implementation of a multisensory system [5]. It combines decisions from neural networks (image analysis) with data analysis from other sensors (accelerometers, sound). Simple prediction weighting in a decision space based on confidence coefficients is impractical due to its sensitivity to techniques and parameter choices. Therefore, we explored various fusion techniques such as Markov Random Fields (MRF) [6] and belief theories (DS) [7]. These fusion methods are necessary because the information available on neural network reliability is inherently imperfect. These imperfections manifest in different forms:

- Uncertainty: Reflects the degree of conformity of information to reality.
- Imprecision: Indicates a lack of accuracy in knowledge, quantitatively.
- Incompleteness: Signifies missing information from a source on certain aspects of the problem.
- Ambiguity: Arises when information leads to two different interpretations.
- Conflict: Emerges when multiple pieces of information yield contradictory conclusions.

Data fusion techniques offer a theoretical framework for processing all types of data and addressing some of the imperfections associated with CNN. However, implementing these techniques entails several steps:

- Information modeling: This step requires selecting a formalism to represent the information to be merged, including their imprecision and uncertainty.
- Data combination: This involves processing sensor information according to predefined rules for information fusion.
- Decision-making: In this step, the system selects a decision from among all proposals formulated by the system.

Our system focuses on detecting five defects: defective fasteners, missing nuts on fishplates, surface defects on the rail tread, and broken rails or fishplates. One challenge the system tackles is the neural networks' confusion between defects and normal situations. For instance, rail breaks are often confused with rail joints due to their similar characteristic: a discontinuity in the rail tread (Figure 1). This shared characteristic renders rail defect identification by deep neural networks nearly impossible. However, detecting contextual elements helps resolve this ambiguity by searching for data, either in the image or from other sensors, to differentiate between the two scenarios. In our example, the sought-after contextual elements are fishplates or braids, which are present in rail joints but absent in the presence of breaks.

Figure 1: *Example of a confusing defect, from left to right: a rail break and a joint*.

Therefore, in our example, the presence of a break will be confirmed by the processing chain if the adjacent areas to the location of the defect show an absence of fishplates and/or braids. In summary, our contribution revolves around the combination of CNN and a fusion model, the core of the forthcoming track monitoring system, for detecting multiple defects. This article is structured as follows:

- In paragraph 2, we will present the state of the art.
- Paragraph 3 outlines the system architecture.
- Paragraph 4 presents the obtained results, which will be discussed in paragraph 5.
- Finally, in paragraph 6, we will conclude this study.

2. State of the Art

Over the past few decades, track monitoring has benefited from technological advances in AI. These new techniques have significantly improved reliability and precision in defect detection [8]. Consequently, defect detection on rails has transitioned from manual visual analysis by agents to automated methods, either based on vision systems [9] or employing alternative methods such as ultrasound or electromagnetism. Traditional methods quickly became inadequate for modern surveillance needs. The new surveillance paradigm aims to automate monitoring and equip commercial trains for regular, real-time infrastructure monitoring. Among the emerging techniques, electromagnetic heating analysis of rails, though yielding promising results, is restricted by an inspection speed of less than 80 km/h [10]. Alternating Current Field Measurement (ACFM) is another method based on electromagnetism, capable of detecting surface or internal defects by analyzing magnetic field deformations, with a maximum analysis speed of 220 km/h on test benches [11]. However, this speed decreases significantly when deployed on commercial trains due to challenges in maintaining a constant distance between sensors and the rail [12]. Magnetic Flux Testing (MFL), which measures the magnetic flux to identify defects, is sensitive to probe placement and allows analysis at a traffic speed of 80 km/h. In the past decade, automatic inspection systems for railway infrastructure using AI techniques have primarily focused on controlling elements such as:

- Fasteners securing the rail to sleepers, a complex problem due to the noisy environment and occlusions caused by erosion, dust, motion blur, and debris.
- Rail breaks.
- Surface defects.

Early automated visual defect detection methods employed signal processing techniques with image decompositions using filters and wavelet transformations for relevant information extraction [13]. Technological advancements in speed have facilitated their adoption in industrial systems [14]. These systems often utilize linear cameras with high acquisition

frequencies. Initial crack and undulation detection systems employed contour detection and various filters to extract patterns on rail treads. Subsequently, detection expanded to include fasteners, nuts, ballast, etc. Machine learning (ML) techniques, initially focused on classification and segmentation, evolved with the introduction of neural network (ANN) models for defect detection. Convolutional neural networks (CNN) have gradually improved performance in terms of precision, recall, and computational efficiency. However, limitations persist, either in the number of defects detected or due to complex processing chains involving prior image processing.

Today, the maturity of CNN significantly enhances defect detection at low acquisition costs, as demonstrated by [15] in tunnel inspection. Compared to other methods like Support Vector Machine (SVM) or Latent Dirichlet Allocation (LDA), CNN exhibits superior accuracy and detection rates [16]. [17] employed Multitask Learning (MTL) to detect sleepers and ties, achieving promising results. [18] evaluated the effect of convolutional network depth on accuracy in large-scale image recognition. [19] utilized a Faster R-CNN network for attachment detection, comprising deep convolutional and R-CNN classifier modules, offering a comprehensive approach to object detection.

3. System Architecture

The core of our processing chain comprises CNN and a fusion model, serving as the cornerstone of the multi-sensor railway infrastructure monitoring system. This innovative system must address several challenges: 1) Implementing an autonomous and reliable defect detection system. 2) Developing a system resilient to failures; 3) Ensuring low energy consumption not exceeding 1000 W; 4) Enabling real-time processing of defects and 5) Ensuring compatibility for installation on board all types of trains. The architecture of this new monitoring system is depicted in Figure 2. The diagram highlights the central role played by the chain combining CNN and the fusion model.

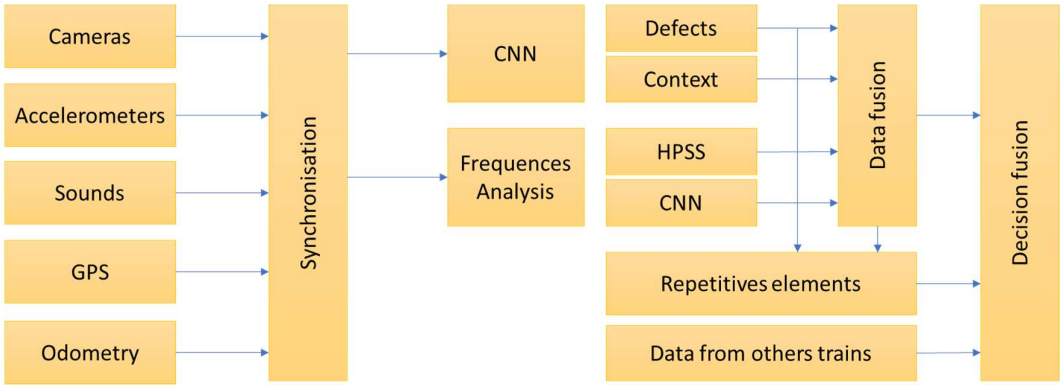

Figure 2: *Architecture of the analysis system.*

The complete processing chain consists of 5 blocks:

- The first block incorporates sensors such as cameras, accelerometers, GPS (to locate defects), and train speed information, essential for data synchronization.
- The second block ensures coherence and timestamping between data frames from the first block.
- The third block, detailed in this article, analyzes, and merges the data.

- The fourth block identifies repetitive elements, specifically rail joints positioned at regular intervals of 12, 18, and 36 meters. This block distinguishes disturbances originating from the rails from defects present on the wheels.
- The fifth block consolidates all decisions from blocks 3 and 4 with detections made by other trains operating on the same line and track, employing a Bond Graph (BG) type graph model.

3.1 Main processing chain

Following this, our focus shifts to the processing of images and sounds, the only elements for which we have actual data. For this segment, we have reinterpreted the UberNet architecture originally developed by [20]. This architecture employs multiple deep artificial networks simultaneously analyzing the same image (see Figure 2).

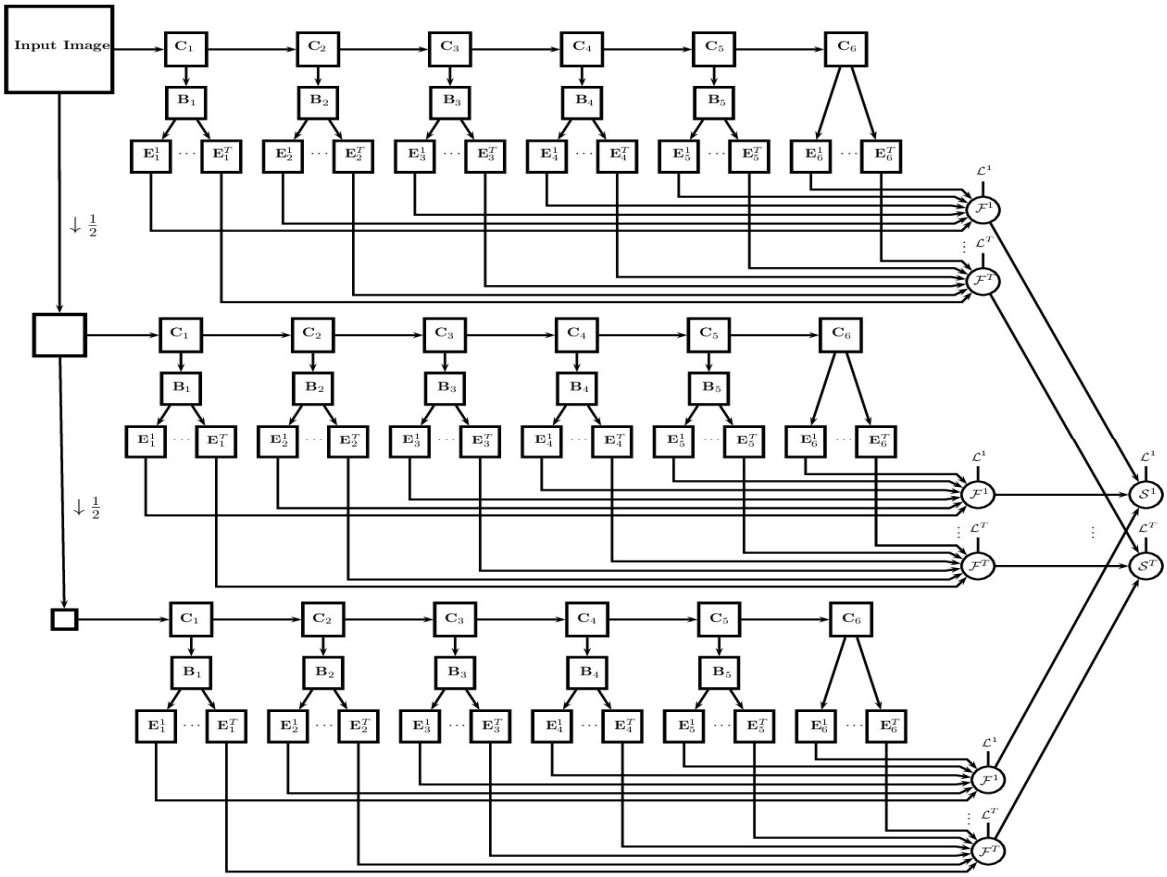

Figure 3: *UberNet architecture [20].*

In our implemented architecture, we utilize:

- Two CNN: one dedicated to defect detection and the other to the detection of contextual elements, acceleration analysis using power density and sound analysis using the HPSS method [21].
- A data fusion module, specifically the belief theory developed by Shafer (1996).
- Each CNN specializes in detecting a specific type of object: defects or context elements.

The concept underlying the fusion model's use is its modularity, structured around three components:

1. Information modeling: This entails selecting a formalism representing the information to be merged, including its imprecision and uncertainty.
2. Combination of information: This involves processing all sensor data according to predefined rules set by the operator.
3. Decision making: The objective is to select a decision among all proposals made by the data analysis modules (CNN, Fourier transform, etc.) dedicated to defect detection.

Therefore, the fusion component, situated at the output of the networks, amalgamates all information from decisions made by the networks or other sources (e.g., accelerometers, microphones). In the intricate landscape of fusion methods, probabilities are often crucial for interpreting stochastic measurements and for dynamic filtering. Fuzzy logic [22] is another tool for interpreting observations, but it only characterizes imprecise measurements. The Kalman filter [23], used in data fusion processes for controlling dynamic systems like robotics or autonomous cars, estimates the system's state from a past state at time t while measuring the differences between the prediction and this new state. However, belief functions, as defined by Dempster Shafer theory, offer a powerful framework for developing data fusion processes, dealing effectively with uncertainty and imprecision.

3.2 Choice of deep neural network

In recent years, CNN have seen significant advancements, especially in terms of inference speed and compactness, aligning well with our requirements:

- Inference times compatible with commercial train speeds, approximately 160 km/h.
- Compact and installable models in embedded architectures composed of FPGA processors and ARM processors.
- A sustainable network model with strong support from the AI community.

The Yolo CNN model [24] best suits our problem. This network excels in detecting areas of interest and labeling detections. Accuracy rates in various studies range from 77% for R-CNN and SSD-type models to over 90% for the latest versions of Yolo architecture [25]. The Yolo architecture comprises three functionally distinct parts: backbone, neck, and head (see Figure 4).

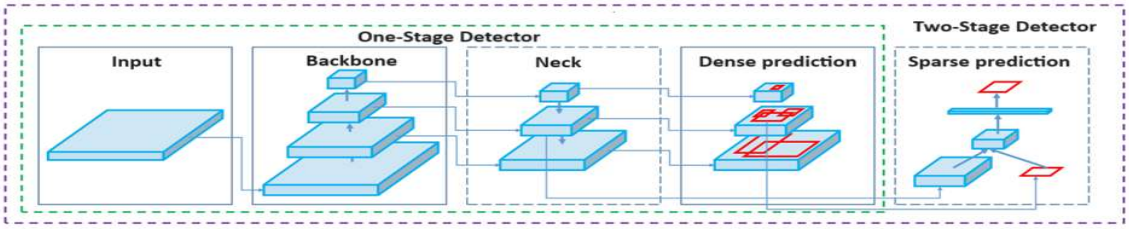

Figure 4: *Architecture of a "one pass" network presented in the form of Backbone, Neck and "Dense Prediction" (head). The input image is provided to the first layer of the backbone, and the head part returns the detections in the form of bounding boxes [26].*

- Backbone − This is a convolutional neural network that aggregates and trains image features at different granularities.

- Neck - This element applies a series of layers to combine the features of an image to pass them to the prediction.
- Head - This element consumes the characteristics provided by the preceding element and undertakes the segmentation and classification steps.

The first version of YOLO was introduced by [24]. In 2015 and provided a breakthrough in real-time object detection. YOLOv1 is a single-stage object detector with fast inference speed and acceptable accuracy compared to two-stage methods of the time. YOLOv2, also called YOLO9000, was proposed a year later to improve detection accuracy by applying the anchor box concept. In 2016, further improvements were made in YOLOv3 with a new Darknet53 backbone and the ability to detect objects at three different scales using Feature Pyramid Network (FPN) as a neck model. [26] introduced YOLOv4 in 2020. YOLOv4 improved the performance of its predecessor YOLOv3 using a new backbone, CSPDarknet53 by adding Spatial Pyramid Pooling (SPP), Path Aggregation Network (PAN) and introducing the method of increasing mosaic data. The YOLO v5 model was introduced for the first time in 2015. This model is positioned from the start on greater speed of execution while maintaining a precision close to previous models. Mandriota compared three models deployed on three devices, namely Jetson Nano, Nvidia GTX 1660 Ti and Nvidia Tesla T4 (table 1). Two parameters make it possible to determine the model best suited to our case, these are the average precision (mAP) and the capacity of the model to process a video, expressed in number of images per second (fps) (table 1).

Table 1: *Comparison between YOLO and MobileNet SSD models on custom dataset [27].*

Model	mAP	FPS		
		Tesla T4	1660	Jetson Nano
YOLOV3	54.3	80	21	8
YOLO V5s	37.6	100	28	15
MobileNet-SSD	33.7	94	26	15

Keeping in mind the criteria mentioned above, we deduce from these results that the YOLO v5s model seems to be the model best suited to our use case, with optimal values of precision and speed. We note that the MobileNet-SSD V2 network offers a speed like to YOLO v5s but its precision is lower.

3.3 The fusion model – belief function theory

Dempster Shafer's theory makes it possible to represent the imprecision and uncertainty introduced by data sources (CNN, sounds, acceleration). This notion of belief proves particularly effective when combining different points of view. It is known for its effectiveness in multiple applications such as image classification or decision-making. The main advantage of this theory is to assign a confidence level, designated by a mass, to the different classes predicted by neural networks while considering the ignorance we have about the information. This theory makes it possible to represent both the imprecision and the uncertainty of a measurement using three functions: mass function, plausibility, and belief. We are interested in the decision taken by a network, identified by a class denoted C, belonging to the set $E = \{C_1, \dots, C_p\}$, here called the discernment framework. A source is supposed to hold at a time t, based on a body of knowledge, an opinion characterized by a degree of belief in each hypothesis of the type "class C belongs to the set A", where A is by definition a subset of set E

[28]. These degrees of belief can be described by a mass function m^E satisfying the following relationship:

$$\sum_{A \subseteq E} m^E(A) = 1 \quad [1]$$

The mass functions that this model manipulates are defined on all subsets of E, and not simply on singletons (contrary to probabilistic and possibilist theories). Two other functions make it possible to represent the information sought: the belief function (bel) and plausibility (pl). The belief or credibility function (bel) represents the degree of confidence in the hypothesis "$C \in A$", considering the masses of belief assigned to all the hypotheses involving A. It is defined by:

$$bel(A) = \sum_{B \subseteq A, B \neq 0} m^E(B) = 1 \quad [2]$$

The plausibility function (pl) represents the maximum degree of belief that can potentially be attributed to the hypothesis "$C \in A$" conditional on obtaining new information: this is the maximum confidence that one can have for the hypothesis. It is defined as follows:

$$pl(A) = \sum_{B \subseteq A, B \neq 0} m^E(B) \quad [3]$$

$$pl(A) = 1 - m^E(\emptyset) - bel(\bar{A}) \quad [4]$$

$\forall A \subseteq E$

Or $\bar{A}$ is the complementary subset of A in E.

The mass $m(A)$ of a given defect expresses the proportion of all available evidence affirming that the defect detected by the network corresponds to the original class and not to another class. The value $m(A)$ assigned by a neural network to a defect does not only concern this defect and does not provide any information on the other defects identified by this same source, but each defect also having, by definition, its own mass. From the value of the mass of a defect, we can define a probability interval. This interval contains the precise value of the probability of identifying the defect, it is bounded by the belief (bel) and plausibility (P) functions:

$$bel(A) \leq P(A) \leq pl(A) \quad [5]$$

Two other functions, called commonality (q) and implicability (b), may be of practical interest in computing the combination of belief structures. These functions are defined as follows:

$$q(A) = \sum_{B \supseteq A} m^E(B) \quad [6]$$

$$b(A) = \sum_{B \subseteq A} m^E(B) \quad [7]$$

This modeling allows, with the use of compound hypotheses, to precisely express the impossibility of dissociating several hypotheses [29]. However, the main interest of Dempster Shafer's theory is information combination. In the case of using several neural networks, it is interesting to combine the knowledge of each source in order to extract an overall knowledge of the defect and to propose a decision to an expert. Let us assume that we have two mass functions m_1^E from m_2^E two distinct neural networks S_1 and S_2. The conjunctive combination is defined by:

$$(m_1^E \oplus m_2^E)(C) = \sum_{A \cap B=C} \frac{m_1^E(A) m_2^E(B)}{1 - m^E(\emptyset)} \quad [8]$$

$$(m_1^E \cap m_2^E)(Y) = \sum_{A \cap B} m_1^E(A) m_2^E(B) \quad [9]$$

$\forall C \subseteq E$

Where $m^E(\emptyset)$ is the conflict coefficient between the two sources. This conflict can have two different origins: either the sources are unreliable, or the information they provide relates to different phenomena. In the case where there is uncertainty about the reliability of the sources, we can use an operator that is more cautious than the conjunctive operator defined above, but which provides less precise information than each of the sources. This operator is modeled as follows:

$$(m_1^E \cup m_2^E)(C) = \sum_{A \cup B} m_1^E(A) m_2^E(B) \quad [10]$$

$\forall C \subseteq E$

Which amounts to applying Dempster's combination rule, expressed by equations (11) and (12):

$$m_{1,2} = \begin{cases} \sum_{B \cap C=A} m_1(B) m_2(C) \text{ si } A \neq \emptyset \\ 0 \text{ si } A = \emptyset \end{cases} \quad [11]$$

This rule can be expressed in its standardized form:

$$m_{1,2} = \begin{cases} \sum_{B \cap C=A} \frac{m_1(B) m_2(C)}{1-K} \text{ si } A \neq \emptyset \\ 0 \text{ si } A = \emptyset \end{cases} \quad [12]$$

With

$$k = \sum_{B \cap C=\emptyset} m_1(B) m_2(C) \quad [13]$$

[30] propose a method which seems more adapted to the data we have, it uses the results obtained via the confusion matrix. This method is based on the reliability rate of a source, it is expressed as follows (eq 14):

$$\tau^i(\{C_i\}) = 1 - \sqrt{\frac{1}{2} \sum_{k=1}^{K} \left(\frac{n_k}{n_j} - \delta_{k,l} \right)^2} \quad [14]$$

With
n_k, representing the number of elements classified in C_i;
n_l, representing the number of type elements C_i ;
$\delta_{k,l}$, being a parameter equal to 1 for the elements of the diagonal of the confusion matrix, or $k = l$; in other cases, this parameter is equal to 0.

We can estimate the level of confidence in the presence of a defect knowing that the truth is framed by the plausibility and belief functions. However, the most used criterion is that of the maximum pignistic probability defined by [31]. This criterion is based on the fact that decision-making consists of choosing the measure for which the expected cost is the lowest. However, in the theory of belief functions we manipulate mass functions and not probability

functions, [31] suggests working with a particular distribution, denoted BetP, obtained by distributing the belief mass $m^E(A)$ equally between the elements of A:

$$BetP(c_i) = \sum_{A \subseteq E} \frac{m^E(A)}{(1 - m^E(\emptyset))|A|}, \ \forall c_i \in E \quad [15]$$

Where |A|is the cardinal of A.

Like any application of a fusion theory, we must consider that the system considered is not an ideal, because one of the neural networks may be absent or defective. In order to model this fact, we assume that the failed component follows a Poisson process with a failure rate denoted λ.

Either:
$X = 1$ if the component fails or is absent in the meantime $[\ 0, t\ ]$;
$X = 0$ for other cases.
The lifespan W of the component follows an exponential law of expectation $\frac{1}{\lambda}$ therefore:

$$V = \lambda W \sim \varepsilon(1) \quad [16]$$

The three variables mentioned above are linked by the relationship:

$$X = 1 \Leftrightarrow \frac{V}{t} \leq \lambda \quad [17]$$

Suppose we only know that $\lambda \in [\underline{\lambda}, \overline{\lambda}]$

If $\frac{V}{t} \leq \underline{\lambda}$ the component is certainly defective, then:

$$bel(X = 1) = m_x(\{1\}) = 1 - e^{-\underline{\lambda}t} \quad [18]$$

If $\frac{V}{t} > \overline{\lambda}$ the component is certainly defective, then:

$$bel(X = 0) = m_x(\{0\}) = e^{-\overline{\lambda}t} \quad [19]$$

Consequently, the failure or absence of a neural network results in the relationship below:

$$m_x(\{0,1\}) = e^{-\underline{\lambda}t} - e^{-\overline{\lambda}t} \quad [20]$$

The confusion matrices associated with the neural networks make it possible to define the mass functions for each class. Indeed, the learning data being known, and according to the work of [32], it is possible to convert the decisions made by neural networks into a belief function via the confusion matrix. The mass function is defined for a class representing a defect or a context element and we must apply the following rule: The detection of a joint between two rails supposes the detection of the elements of the set composed of fishplates, breaks or braids. The combination of the mass functions of each defect/element amounts to identifying the subset of defects corresponding to the intersection of two sets: all the context elements identified by the secondary networks and all the elements identified by the main network. Let C be a set of classifiers and Ω be the set of defects identified by the networks. The classifier is seen as a function C taking as input an object X from a set of objects P as output of a class $C(x) = \omega_k \in \Omega \cup \{\omega_{K+1}\}$ or by convention ω_{K+1} corresponds to the unknown class.

Each row k of the confusion matrix corresponds to the decision of the network $C(x) = \omega_k$ and each column l of this same matrix corresponds to the reality of the classification ω_l. We also use three rates which allow us to assess the performance of each network:

The recognition rate, noted R;

$$R_i = \frac{\sum_{k=1}^{K} n_{kk}^i}{n^i} \quad [21]$$

The rejection rate, noted T;

$$S_i = \frac{\sum_{k=1}^{K} \sum_{l=1; l \neq k}^{K} n_{kl}^i}{n^i} \quad [22]$$

And the substitution rate, noted S.

$$T_i = \frac{n_{(K+1)}^i}{n^i} \quad [23]$$

For all the rates defined above, n^i represents all the elements of one of the classes of the validation dataset or depending on the level of aggregation of the classes, all of the elements to be identified. The rates R, S and T make it possible to calculate two other rates essential for calculating the mass functions, the reliability rate and the rate measuring the unreliability of the network. Thus, the reliability rate of the classifier, without considering rejections, is defined as follows:

$$R_i = \frac{R_i}{1 - T_i} \quad [24]$$

The rate measuring the unreliability of a network is defined by the following relationship:

$$U_i = \frac{S_i}{1 - T_i} \quad [25]$$

Consequently, the mass function is defined for each class by the following relation:

$$m_i(\{\omega_l\}) = \frac{n_{kl}^i}{n_k^i} = R_i \quad [26]$$

The introduction into the calculation of the mass function of a contextual weakening makes it possible to consider that the reliability of one of the sources can vary depending on the truth of the object to be recognized. A simple method, proposed by [30] consists of using its expression by the disjunctive combination rule.

$$m^{\alpha} = m \cup_{l=1}^{K} \{\omega_l\} R_i [\omega_l] \quad [27]$$

In the case where a source of information, from which the belief function comes, is not completely reliable, it is possible to introduce a weakening variable. Let α be the attenuation variable associated with the source:

$$\alpha = \frac{n_{kk}}{n^i} \quad [28]$$

The coefficient α represents the knowledge of the reliability of the source. Thus, a weakened mass function, denoted m_α, can be deduced from the initial mass m via the following relationships:

$$m_\alpha(A) = \alpha m(A)\ \forall A \in \Omega, A \neq \Omega \quad [29]$$

$$m_\alpha(A) = 1 - \alpha + \alpha m(\Omega) \quad [30]$$

A belief function can represent several types of knowledge and thereby constitutes a rich and flexible framework for the representation of uncertain information. It is interesting to treat, via this model, the failure of one or more components of our processing chain. In order to simulate this restricted operating mode, we associate with the normal operating mode of a sensor or a layer, denoted s_k, a maximum performance level, denoted u_k, the total failure mode of the sensor, denoted s_0, is associated with a minimum performance level denoted u_0. We assume that the system works when the main component works. By main component, we mean the layer in charge of defect detection. For the sake of simplification, we consider that each component has the same failure rate noted λ. The events that drive components from the $\{Working\}$ state to the $\{Failed\}$ state is independent. In this case, the system being considered homogeneous, its reliability is given by the following relationship:

$$R(T_i) = \sum_{i=k}^{n} C_n^i r(T_i)^i \left(1 - r(T_i)\right)^{n-i} \quad [31]$$

With

$$C_n^i = \frac{n!}{i!(n-i)!} \quad [32]$$

Where $r(T_i)$ is the probability for each component to be in the $\{On\}$ state and T_i is the mission duration. By setting $T_i = 200h$ and taking as a pessimistic hypothesis a reliability of 0.95 for each layer, the theoretical reliability of the system is established at:

$$R(T_i) = 0{,}9235 \quad [33]$$

4. Results

The networks were trained from a training base comprising 21,000 images, or on average 2,300 images per object. The analysis chain (CNN + fusion model) was tested on 450,000 images. The training base is divided into two groups:

- The group of defects includes missing or defective fasteners, missing nuts, broken rails or fishplates and surface defects,
- The group of contextual elements consists of fishplates, braids, clamping and marking elements.

From the training base, we tested several architectures of Convolutional Networks, the results of this comparison are presented in figure 5. At the end of this comparison, we have chosen the CNN YOLO v5. For each version of this CNN and before inference, we preprocessed the images. This preprocessing consists of re-cutting images, we went from an image with an average size of 1500 x 700 to an image with a standardized size of 750 x 700.

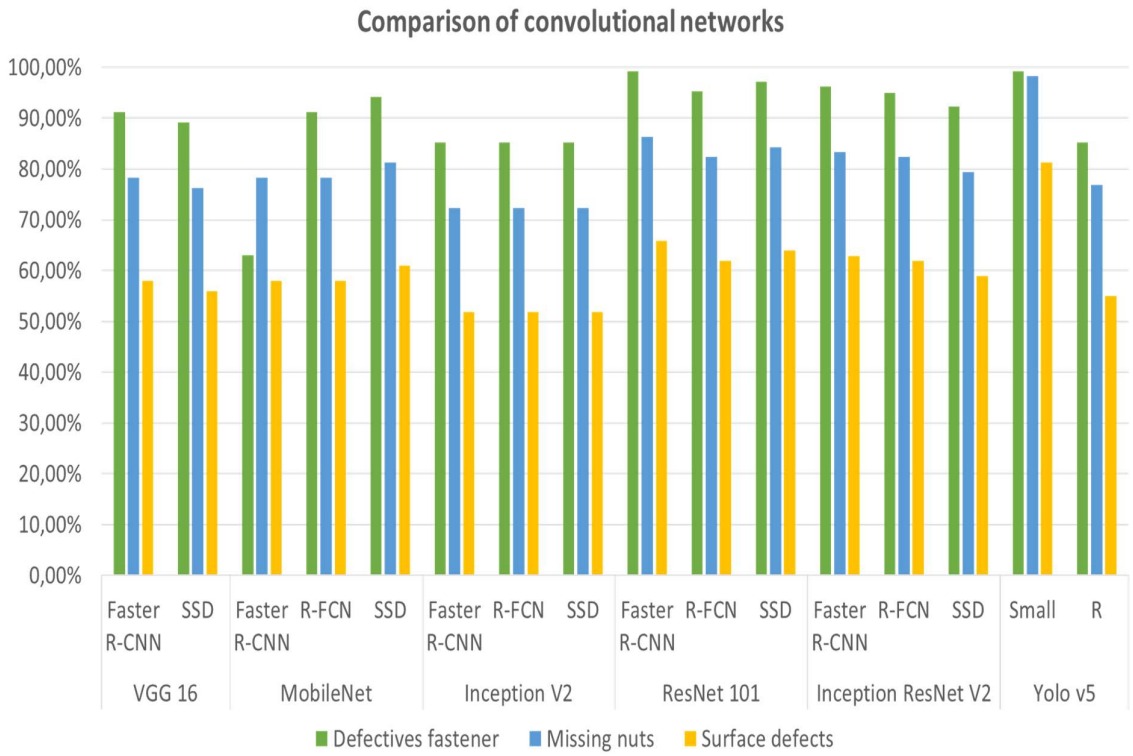

Figure 5: *Choice of convolutional network.*

4.1 YOLOV5 Networks

The results obtained by the Yolo network (Table 1), although good, highlight the difficulty that CNN have in discriminating defects whose characteristics are close to normal situations. Indeed, we observed the following confusions:

- A high number of missing nuts were detected even though they were just holes.
- The joints were systematically confused with rail breaks.
- Surface defects were mistaken for oil stains or vegetation.

Concerning the detection of defective fasteners, we notice a difference between the number of missing fasteners detected by our network and the number of fasteners reported by an operator. This difference is due to a management rule specific to the French infrastructure manager, SNCF Reseau, the missing fastener defect is validated if at least three consecutive fasteners are absent. Once we selected the YoloV5 model, we tested the small and medium variants. For each test we carried out a first inference without pre-processing the images then in a second inference with pre-processing. The results show that the models perform better when inferring raw images (figure 6) than when inferring preprocessed images (figure 7).

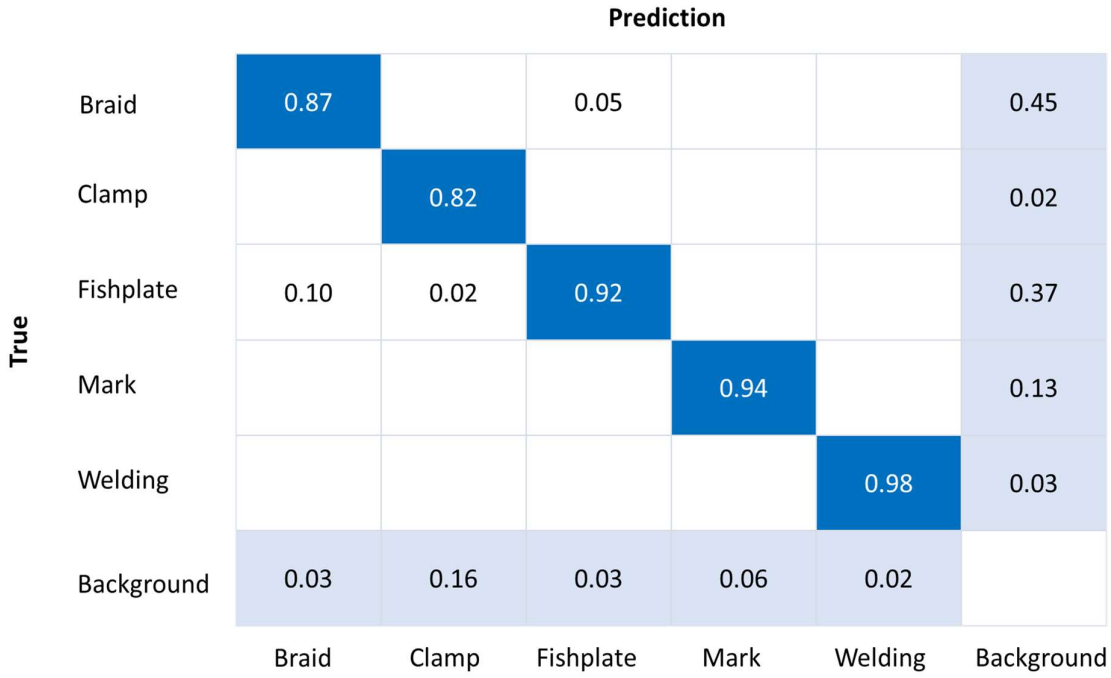

Figure 6: *Yolov5 small confusion matrix without pre-processing.*

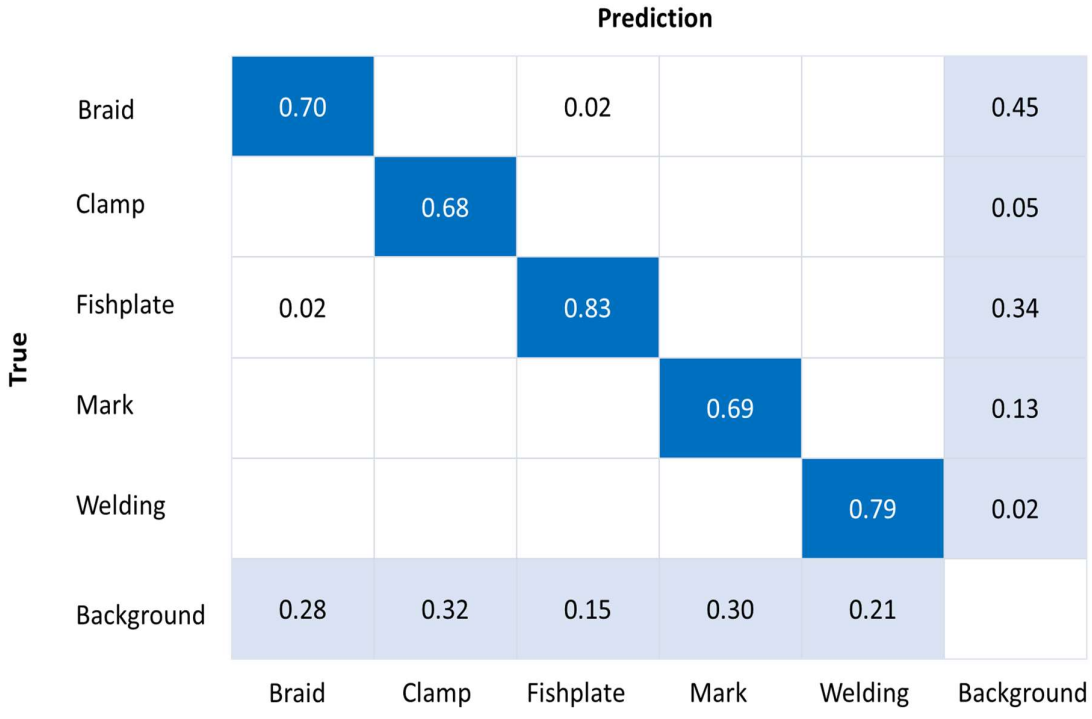

Figure 7: *Yolov5 small confusion matrix with pre-processing.*

During tests on our dataset the YOLO V5 small model obtained better results than the YOLO R variant (table 2).

Table 2: *Comparison between YOLO v5 small and YOLO R.*

	YOLO v5small				YOLO R			
	Precision	**Recall**	**mAP@.5**	**mAP@.5-.95**	**Precision**	**Recall**	**mAP@.5**	**mAP@.5-.95**
Defective fasteners	0.992	0.995	0.993	0.789	0.852	0.954	0.945	0.615
Missing nuts	0.983	0.991	0.987	0.628	0.769	0.955	0.952	0.445
Breakups	0.998	0.455	0.5	0.253	-	-	-	-
Surface defects	0.812	0.845	0.894	0.531	0.55	0.918	0.956	0.409
Clamps	0.983	1	0.994	0.663	0.704	0.867	0.846	0.469
Fishplate	0.999	0.999	0.995	0.923	0.73	0.988	0.982	0.647
Braid	0.953	0.985	0.98	0.705	0.47	0.984	0.946	0.509
Seal	-	-	-	-	0.66	0.875	0.827	0.30

Figure 8 illustrates the differences in decisions made by three systems: The commercial monitoring system, used by the French infrastructure manager, SNCF Réseau, this new analysis system, and the operator. In cases 1 and 2, the current monitoring system detects a fishplate breakage even though the operator and our new system made the right decision: no fault. Case 3 shows an interesting case, the use of a CES clamp used to consolidate a critical rail defect. in this case both automated systems detect a missing nut. The addition removes this ambiguity and reclassifies this defect.

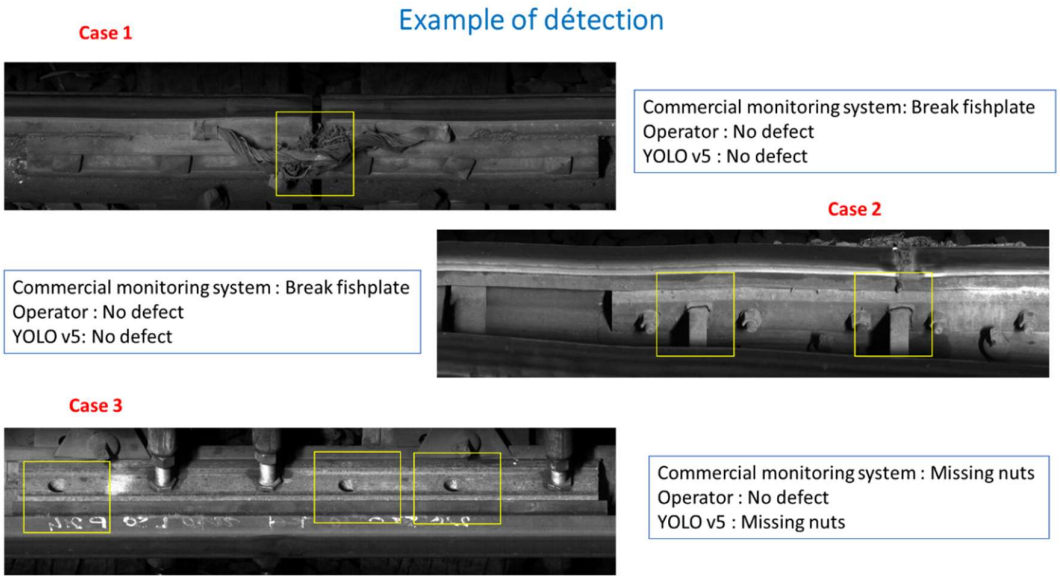

Figure 8: *Choice of convolutional network.*

4.2 Evaluation of the fusion model

The distinction between a break and a joint being almost impossible by CNN, the theory of belief functions makes it possible to consider the disjunction of these two classes without introducing arbitrary information forcing their separation. The type of architecture studied consists of three classifiers:

- The classifier C_1 for defect detection,
- The classifier C_2 for contextual elements,
- The classifier C_3 for detecting joint-specific sounds.

Let the set $\Omega = \{C_1, C_2, C_3\}$ or each C_i designate one of the three classifiers and $m_j (j = \{C_1, C_2, C_3\})$, the mass function defined for one of the three classifiers. Each neural network is specialized in the following classifications:

Table 3: *Affiliation of classes to sources.*

Classifiers	Classes detected
C_1	Defective fasteners, defective nuts, broken rails and fishplates, surface defects.
C_2	Fishplates, braids.
C_3	Joints, ambient noise.

In the case of probabilistic classifiers, the results obtained in the same discernment framework from two independent neural networks and expressed by the two mass functions m_1 and m_2 can be combined in order to obtain a common base assignment noted $m_{1,2}$. In general, results can be combined in different ways, some of which may consider the reliability of the sources. This first approach is based on the following two practical and theoretical arguments:

- First argument: The rule of prudence and other t-norm based rules cannot be directly applied to probabilities, because they are dogmatic belief functions.
- Second argument: The outputs of the probabilistic classifier are interpreted as betting probabilities, whereas any belief function whose graphical probability distribution matches the classifier output can be considered consistent with that output.

Table Table , below, gives for each defect, the evidence as well as the sources used.

Table 4: *Rules of evidence.*

Defects/elements	Merge Rules
Joint versus rail break	This element is identified by the classifier C_1 in association with the detection of a fishplate or braid (Classifier C_2) and a specific sound (Classifier C_3). The absence of one of the elements cited directs the system's decision towards a rupture. Let the set A be composed of these elements {breaks, fishplates, braids, joint}.
Surface defects	This type of defect is identified by the classifier C_1 and it is confirmed by the frequency detected by the classifier C_3. Consider the set B comprising the following singletons {Surface defects, sounds}.

Fishplate break	This type of defect is identified by the classifier C_1 provided that a fishplate is detected via the classifier C_2. Consider the set D comprising the following singletons {breaks, fishplates}.

We calculated the confidence and weakening rates for each defect or context elements (Table 5). The defects identified by the classifier C_1 have the lowest confidence rate despite a weakening rate almost equivalent to those calculated for the other classifiers. This result can be explained by the low number of defects present in the dataset compared to the number of false positives detected by the monitoring system used on the national rail network.

Table 5: *Mass function and attenuation for each source.*

Classifier	Defects (Classes)	Confidence rate	Attenuation rate	Mass function $m(\Phi)$	Mass function $m(A)$
C_1	Defective fasteners	0.163	1,000	0.0569	0.943
	Missing nuts	0.359	0.760		
	Break	0.006	0.375		
	Surface defects	0.123	0.933		
	Others	0.996	0.818		
C_2	Fishplates	0.981	0.946	0.0140	0.986
	Braid	0.970	0.966		
	Others	0.672	0.778		
C_3	Joint	0.856	0.941	0.1086	0.891
	Others	0.954	0.841		

In order to measure the differences in behavior of the classifiers we propose to use the reliability rate R_i of the classifier C_i. These assignments only consider the reliability of the classifiers when a class is chosen. Before obtaining the information from the classifier C_i, all classification options are plausible. Then, when the classifier C_i gradually assigns elements to classes ω_k, the set of elements that do not correspond to one class ω_k becomes more and more restricted. The size of this set depends on the reliability of the classifier. A variant of proposition (32) taking this fact into account is proposed below:

$$\begin{aligned} &m_i = 2^{\Omega} \rightarrow [0,1] \\ &\{\omega_k\} \rightarrow R_i \\ &\Omega \setminus \omega_k \rightarrow U_i = 1 - R_i \end{aligned} \quad [34]$$

The reliability of the classifier also influences the degree of belief of the class $\{\omega_k\}$. If the classifier is unreliable, we are tempted to think that the answer given for an element is not the correct one. In this case, the introduction of this classifier in the data fusion part will increase the conflicts, which is why we chose to generalize the flexibility to our problem and therefore the conditions described in (32). Table Tablepresents the recognition and substitution rates of all the classifiers considering the assignments described in equation 31.

Table 6: *Calculation of classifier reliability rates.*

Classifiers	R_i Recognition rate	S_i Substitution rate	T_i Rejection rate
C_1	0.943	0.0569	0.000
C_2	0.986	0.0140	0.000
C_3	0.891	0.1086	0.000

In view of the results obtained, we see that the rejection rate for all the classifiers is zero. All the elements have been distributed into one of the classes. These results allowed us to categorize the networks by dividing them into four categories (Figure 9).

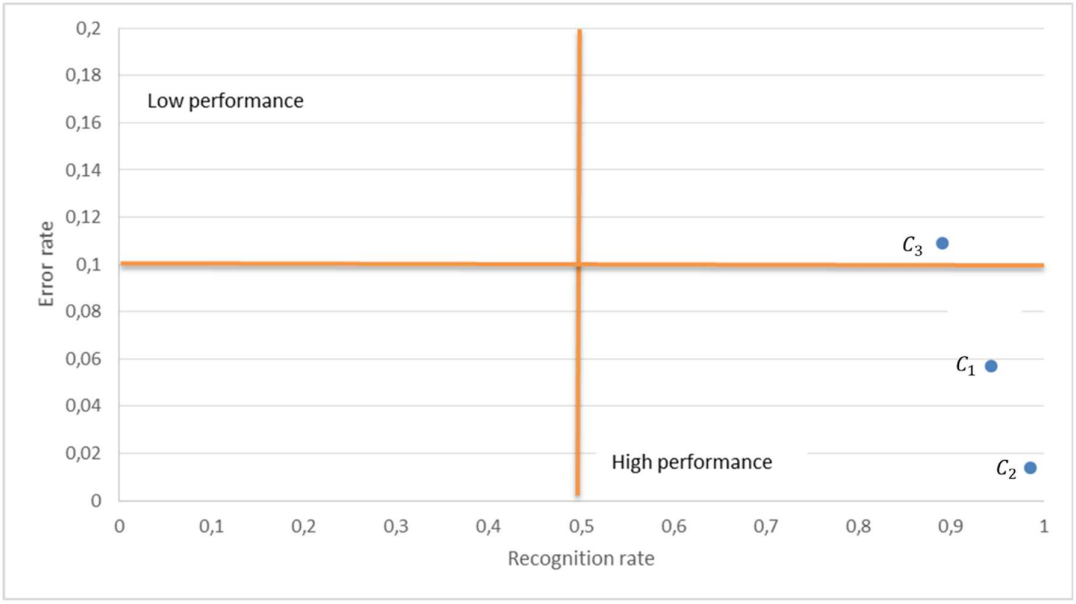

Figure 9: *Representation of classifier performance.*

All networks are grouped into the "high performance" category. We now consider a set of 3 networks C_i, with $i \in \{1,..,3\}$ where each element x is assigned to one of the classes belonging to the set $\Omega = \{\omega_1,..., \omega_k\}$. This set is assumed to be common to all classifiers. As classifiers can select a set of classes, the rejection class is equivalent to a decision for the set of classes Ω. Consequently, the rejection will now be denoted $C_i(x) = \Omega$, instead of $C_i(x) = \omega_{K+1}$. In order to illustrate this mechanism, consider the combination which identifies the rail joints, i.e. A, the set of classes $\{\omega_{Rupture}, \omega_{Fishplates}, \omega_{Tresse}, \omega_{Sons}\}$ coming from three different sources C_1, C_2 and C_3, with C_1 = {Breaks}, C_2= {fishplates, Braids}, and C_3 = {Sounds}. Depending on the difficulty of the recognition task, each sensor and associated classifier can recognize either a defect, a defect class for example ($\omega_{Rupture}, \omega_{Fishplates}, \omega_{Sons}$) or a contextual element. In case of high uncertainty, the system can also select the reject option, which is like to choose the option in our merger model, the entire universe Ω. The hierarchical decision space corresponding to this case is represented by 10.

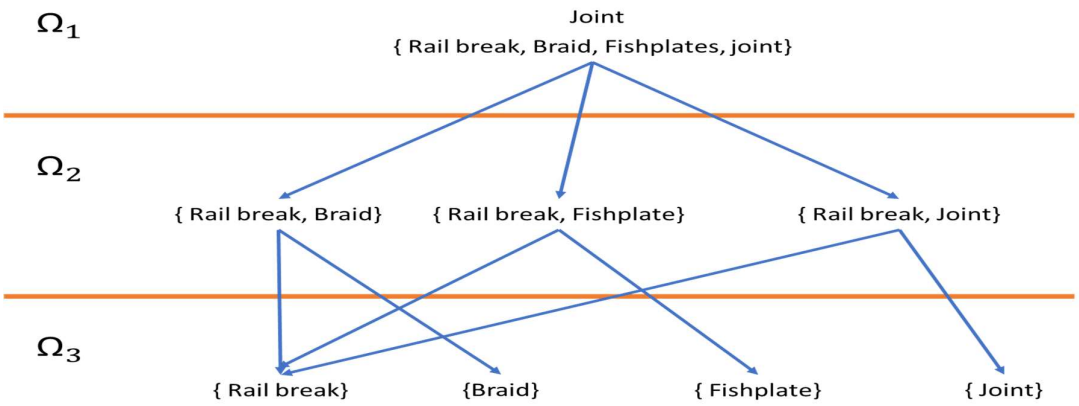

Figure 10: *Hierarchy of combinations for joint detection.*

It should be noted that the classifier detecting breaks mainly detects joints, real rail breaks will be detected if one of the following elements {braid and Fishplate or sounds} are not present.

Table 7: *Calculation: mass functions, plausibility functions, pignistic probabilities of elements or defects.*

Defects	m(Φ) weakened	m(A) weakened	Pl(A) Weakened	Bel(A) weakened	BetP(A) Weakened
Fishplate	0.104	0.896	0.104	0.896	0.930
Braid	0.068	0.932	0.068	0.932	0.955
Rail break	0.859	0.141	0.859	0.141	0.313
Surface defects	0.129	0.871	0.129	0.871	0.897
Joint	0.114	0.886	0.114	0.886	0.943
Ambient noises	0.292	0.708	0.292	0.708	0.854

The result obtained for the detection of rail joints by data fusion shows that the pignistic probability is well below 50% compared to the 90% obtained for the context elements (Table). This result is explained by the low number of rail breaks (less than 40 breaks were detected across the entire national network) present on the national rail network and identified by the automated detection system in relation to the mass of the joints misclassified as a breakup. Another probable cause is that the systems were trained to detect a break, but the detection of joints is a false positive. The application of the weakening coefficient accentuates this difference and brings the confidence level down to around 10%. Each layer of the UberNet network corresponds to a performance level denoted u_i as well as an operating mode denoted s_j. Let us now study the impact of the failure of one or more components. Thus, in the fusion chain leading to the recognition of a joint, we have alternately inhibited the sensors allowing the identification of the following context elements: the fishplates, the braids, the sound specific to the joint and the "rail break". The results are presented in the table below:

Table 8: *Calculates weakened mass functions for each combination of failed components.*

Combination	Defective sensors				
	Level	Fishplate	Braids	Seal	Breakup
Fishplate, breaks	2	0.000	0.584	0.584	0.000
Braids, breaks	2	0.692	0.000	0.692	0.000
Joint(s), ruptures	2	0.354	0.354	0.000	0.000
Fishplate, breaks, joints (sounds)	1	0.000	0.435	0.000	0.000
Braid, breaks, joint(sounds)	1	0.552	0.000	0.000	0.000

First observation, the failure of a sensor identifying the specific context element of sounds leads to the failure of the overall system. Despite the failure of the sensors dedicated to the detection of ribs and braids, we can maintain minimal operation of the system at the cost of a significant increase in the number of false positives. Failure of the main sensor results in complete failure of the device. Let us now consider that the component failure rate is not known precisely but in the form of an interval: $[\underline{\lambda}, \overline{\lambda}] = [0{,}9e^{-3}, 1{,}1e^{-3}]$. As an example, we will use the values of λ proposed by [33]. Applying this distribution relative to the state of the sensors for joint identification gives the following results:

Table 9: *New distributions of mass functions considering the state of sensors.*

Defects	m(A)	Pl(A)	Bel(A)	BetP(A)
Fishplate, break	0.312	0.688	0.312	0.656
Braid, break	0.323	0.677	0.323	0.662
Joint(s), ruptures	0.147	0.853	0.147	0.573
Fishplate, break, joint (sounds)	0.072	0.928	0.072	0.382
Braid, breaks, joint (sounds)	0.076	0.924	0.076	0.384

The integration of this failure makes the system inoperable if we only consider the credal probability. However, considering the pignistic probability makes it possible to maintain acceptable operation of the system.

Table 10*: Calculation: weakened mass functions, weakened plausibility functions, weakened pignistic probabilities of elements or defects.*

Defects	pl(A)	Bel(A)	BetP(A)	Pl(A) Weakened	Bel(A) weakened	BetP(A) Weakened
Fishplates, breaks	0.0495	0.9505	0.7922	0.4157	0.5843	0.7922
Braids, breaks	0.0346	0.9654	0.8462	0.3076	0.6924	0.8462
Joint (sounds), rail breaks	0.1294	0.8706	0.6769	0.6462	0.3538	0.6769
Fishplates, breaks, joints (sounds)	0.0246	0.9754	0.6233	0.5651	0.4349	0.6233
Braid, breaks, joint (sounds)	0.0169	0.9831	0.7014	0.4480	0.5520	0.7014
Joint: {Fishplates, breaks, joint (sounds), braids}	0.0007	0.9993	0.6578	0.5132	0.4868	0.6578

The acceptance phase involved comparing defect detection between SNCF Reseau's current track monitoring system, validated by an expert's decisions, and our new system. Both systems analyze the same lines and tracks (Table 11).

Table 11: *Comparison between current monitoring system and the new monitoring system.*

Line - Track	Current track monitoring system Defects valided by experts		New track monitoring system Defects detected automatically		Gap between systems	
	Defective fastener	Missing nuts	Defective fastener	Missing nuts	Defective fastener	Missing nuts
1000-V1	5		5		0	0
1000-V2	47		56		9	0
180000-V2	8		8		0	0
20000-V1	1		1		0	0
205000-V1	16	26	19	27	3	1
229000-V1	45		49		4	0
229000-V2	35		47		12	0
267000-V1	8		8		0	0
272000-V2	46		53		7	0
295000-V1	2		3		1	0
32000-V2	2		2		0	0
334000-V1	1		3		2	0
340000-V1	23		25		2	0
340000-V2	3		3		0	0
570000-V1	4	3	5	4	1	1
570000-V1BIS	2		2		0	0
570000-V2	2		2	2	0	2
655000-V1	44	1	45	2	1	1
655000-V2	26	2	31	3	5	1
70000-V2	58		66		8	0
750000-V1	32		32	3	0	3
750000-V2	6		6		0	0
752330-V1	9		9		0	0
752330-V2	1		1		0	0
783000-V1	6		8		2	0
798000-UNIQUE	1		2		1	0

The result shows comparable results for the detection of missing nuts and deviations when detecting defective fasteners. This discrepancy is due to internal management rules that are not applied in our processing chain. It is noted that the current system, before the validation made by the experts, produces a considerable number of false positives. Out of 1,500,000 images submitted to the experts, less than 1% are validated.

5. Discussion

The combination of neural networks and a fusion model has yielded outstanding results. The detection rates achieved for each defect over 0.80, reaching as high as 0.95 for defective fasteners and surface defects. Integrating the fusion model for image analysis and incorporating additional information sources (such as sounds) led to a nearly 90% reduction in false positives and a substantial enhancement in rail break detection accuracy, soaring from 0.33% to 95.2% (Figure 11). To further enhance surface defect detection, additional information such as acceleration measurements is required.

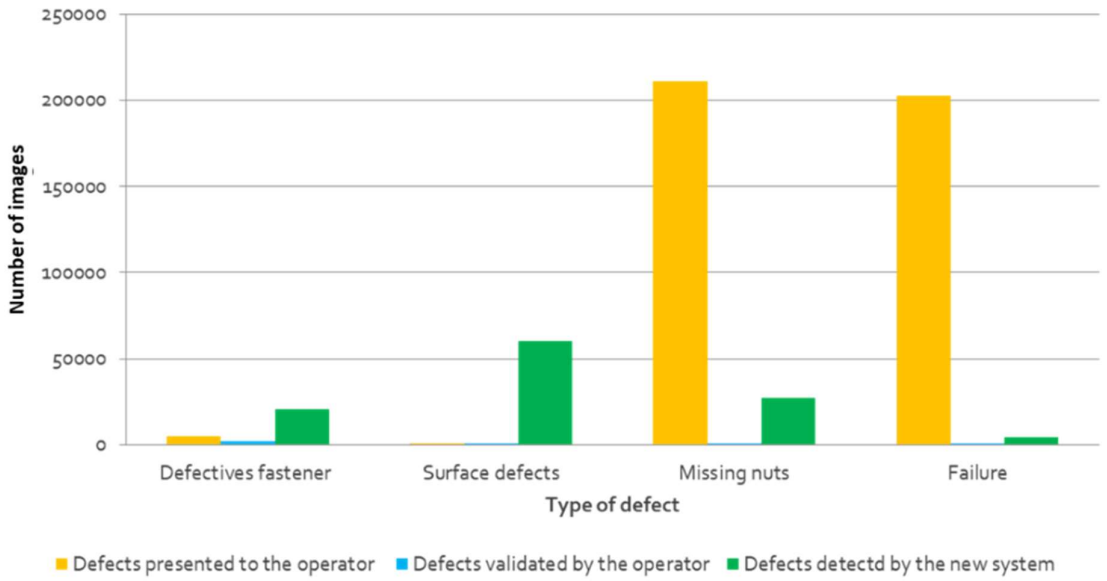

Figure 11: *Results obtained by the new system.*

However, we also observe a degradation in the detection of defective fasteners. This decline stems from a management rule implemented in the surveillance machines' software, which has not been incorporated into our model. According to this rule, a fastener is classified as defective if at least two consecutive defective fasteners are detected. Comparing a multi-defect monitoring system with those focusing solely on single defect detection reveals significant disparities in efficiency and reliability. Single detection systems may exhibit heightened sensitivity to specific defects but often overlook critical issues that can compromise traffic safety. In contrast, our multi-defect system offers broader coverage, identifying a wider range of potential issues. This facilitates proactive maintenance and reduces the risk of major incidents. Despite achieving slightly lower performance than hyper-specialized systems, our versatile system is simpler to implement and maintain. This simplicity provides a crucial advantage in terms of safety and operational efficiency in modern rail networks.

Consequently, the simultaneous analysis of information from various sources, such as images or sounds, has enabled us to eliminate false positives and strengthen the detection of certain defects. Consider the example of fishplate breaks: in conventional systems, the fishplate breakage zone is often confused with the rail junction zone. However, our processing chain, aided by the fusion model, can accurately identify this critical defect. Moreover, the detection of contextual elements has also yielded positive results. For instance, implementing the recognition of paint marks on the rails, which identify the defects under surveillance, has led to an 80% reduction in surface defects. Similarly, for missing nuts, fishplate detection has

reduced false detections by 70%. Consequently, the 95.7% accuracy in detecting missing nuts mostly corresponds to genuine defects. Unfortunately, we were unable to measure the improvement for break detection, as this defect was absent from our test set.

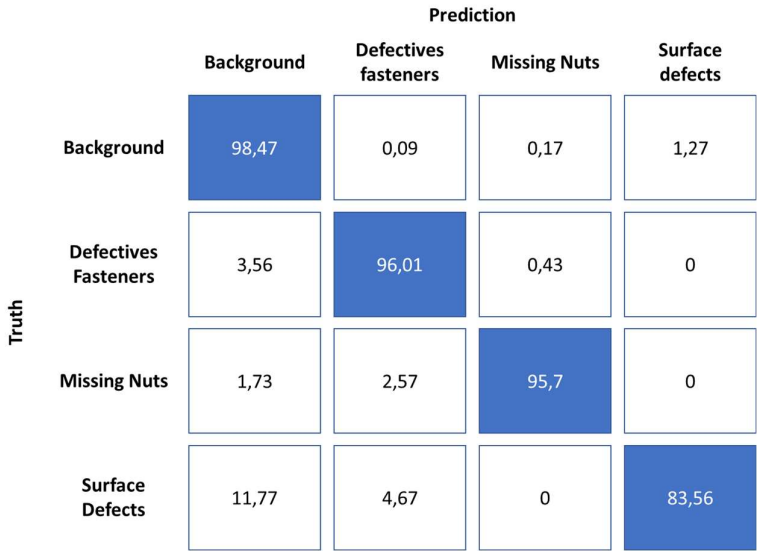

Figure 12: *Confusion matrix obtained by the final system.*

Due to their heterogeneous shapes, surface defect detection is the least effective, although it remains at a satisfactory level. Confusion between real defects and the presence of oil or vegetation stains also contributes to this outcome. The incorporation of accelerometers is expected to significantly decrease the number of false positives. While this defect detection technique has yielded promising results, certain challenges persist. Exploring solutions such as integrating new sources of information (e.g., accelerometers, sounds) is necessary. Additionally, the overall procedure time and image processing duration must be reduced for these methods to be industrialized successfully. In Figure 13, we present some examples of defect detection with annotations. It can be observed that our networks successfully detect all types of annotated defects, including rail and fishplate breaks, defective fasteners, missing nuts, and surface defects.

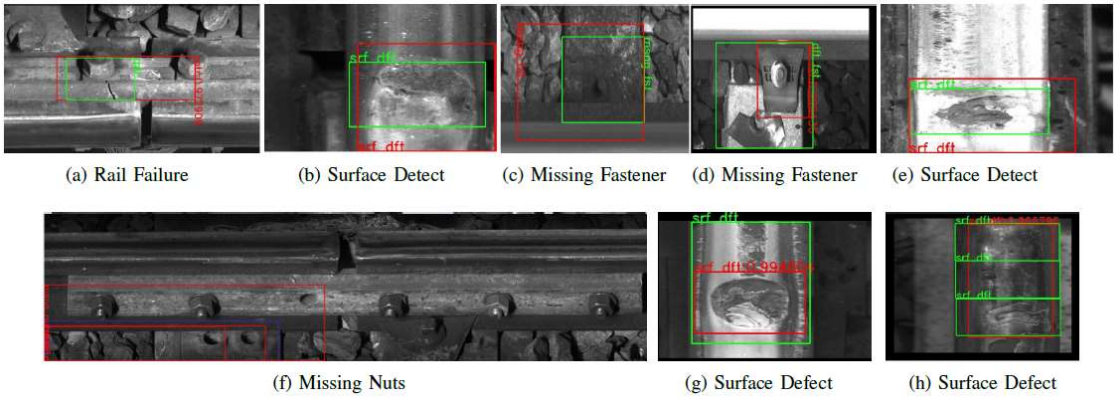

Figure 13: *Examples of detections performed by neural networks on elements of the validation database. The green bounding boxes highlight the annotated areas.*

Belief functions serve as tools for assessing subjective probability, enabling the evaluation of the degree of truth in an expert opinion. By introducing evidence masses, attenuating coefficients for these masses, and utilizing the combination rule, we can process information and arrive at reliable decisions. However, [34] highlights difficulties in applying this theory due to:

- Sensitivity of the combination method to small values of belief masses.
- Significant calculation times required compared to other methods.
- Lack of clear semantics.

Conversely, [35] and [36] view belief functions as a tool that enables reasoning in uncertainty with degrees of belief broad enough to encompass classic probabilistic and possibilist approaches. It also facilitates comparing views, merging criteria to process information from different sources and domains.

6. Conclusion and Perspectives

The results obtained by our architecture demonstrate that the approach combining neural networks and a fusion model is particularly well-suited for pathway monitoring. It effectively distinguishes defects from normal situations, reflecting the reasoning pattern implemented by an agent and reducing false positives generated by conventional image processing techniques. The utilization of belief theory enables the proposal of a robust system resilient to failures or uncertainties while maintaining relevant performance, provided that these failures do not affect the main classifier. Our future work will concentrate on developing the entire processing chain, focusing particularly on:

- Evaluating the latest versions of Yolo, including V8 versions and MobileNetV3 models, as well as the Smets latest networks developed by Google.
- Data synchronization.
- Implementing repetitive elements using Kalman filters.
- Adding a decision fusion model to eliminate the remaining false positives.

References

1. Ruta A, Li Y, Liu X. Robust class similarity measure for traffic sign recognition. IEEE Trans Intell Trans Sys. 2010;11:846-55.

2. Dalal N, Triggs B. Histograms of oriented gradients for human detection. IEEE computer society conference on computer vision and pattern recognition (CVPR'05), San Diego, CA, USA. 2005.

3. Xia Y, Xie F, Jiang Z. Broken railway fastener detection based on adaboost algorithm. International conference on optoelectronics and image processing, Haikou, China, 2010.

4. Wang J, Fu P, Gao RX. Machine vision intelligence for product defect inspection based on deep learning and Hough transform. J Manuf Syst. 2019;51:52-60.

5. Durieu C, Aldon MJ, Meizel D. Multisensory data fusion for localisation in mobile robotics. Trait Signal. 1996;13:143-66.

6. Li SZ. Markov random field modeling in image analysis. (3rdedn), Springer Science & Business Media, Heidelberg, Germany, 2009.

7. Shafer G. A mathematical theory of evidence. Princeton University Press, Princeton, New Jersey. 1976.

8. Micic M, Brajovic L, Lazarevic L, et al. Inspection of RCF rail defects–review of NDT methods. Mech Syst Signal Process. 2023;182:109568.

9. Deutschl E, Gasser C, Niel A, et al. Defect detection on rail surfaces by a vision based system. IEEE Intelligent Vehicles Symposium, Parma, Italy. 2004.

10. Topp D. Recent developments and applications of the ACFM inspection method and ASCM stress measurement method. Non-Destructive Testing Australia. 2005;42:143.

11. Xu P, Chen Y, Liu L, et al. Study on high-speed rail defect detection methods based on ECT, MFL testing and ACFM. Meas. 2023;206:112213.

12. Munoz JC, Marquez FG, Papaelias M. Railroad inspection based on ACFM employing a non-uniform B-spline approach. Mech Syst Signal Process. 2013;40:605-17.

13. Mandriota C, Nitti M, Ancona N, et al. Filter-based feature selection for rail defect detection. Mach Vis Appl. 2004;15:179-85.

14. Ren Z, Fang F, Yan N, et al. State of the art in defect detection based on machine vision. Int J Precis Eng Manuf Green Tech. 2022;9:661-91.

15. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst. 2012;25:1097-105.

16. Jelodar H, Wang Y, Yuan C, et al. Latent Dirichlet Allocation (LDA) and topic modeling: models, applications, a survey. Multimed Tools Appl. 2019;78:15169-211.

17. Gibert X, Patel, VM, Chellappa R. Deep multitask learning for railway track inspection. IEEE Trans Intell Transp Syst. 2016;18:153–64.

18. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 3rd International conference on learning representations, San Diego, CA, USA. 2015.

19. Ren S, He K, Girshick R, et al. Faster r-cnn: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst. 2015;28:91-9

20. Kokkinos I. Ubernet: training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, Hl, USA. 2017.

21. Akaishi N, Yatabe K, Oikawa Y. Harmonic and percussive sound separation based on mixed partial derivative of phase spectrogram. IEEE International conference on acoustics, speech and signal processing, Singapore. 2022.

22. Mateus BC, Farinha JT, Mendes M. Fault detection and prediction for power transformers using fuzzy logic and neural networks. Energies. 2024;17:296.

23. Bertipaglia A, Alirezaei M, Happee R, et al. An unscented kalman filter-informed neural network for vehicle sideslip angle estimation. IEEE Trans Veh Technol. 2024:1-15.

24. Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, Nevada. 2016.

25. Zhang C, Xu D, Zhang L, et al. Rail surface defect detection based on image enhancement and improved YOLOX. Electronics. 2023;12:2672.

26. Bochkovskiy A, Wang CY, Liao HY. Yolov4: Optimal speed and accuracy of object detection. ArXiv Prepr. 2020:1-17.

27. Iyer R, Ringe PS, Bhensdadiya KP. Comparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for real-time mask detection. Artic Int J Res Eng Technol. 2021;8:1156-60.

28. Challa S, Koks D. Bayesian and dempster-shafer fusion. Sadhana. 2004;29:145-74.

29. Germain M, Boucher JM, Benie GB, et al. Multi source evidential fusion based on a new fuzzy statistical appraoch. ISIVC04, Brest, France, 2004.

30. Elouedi Z, Lefevre E, Mercier D. Discountings of a belief function using a confusion matrix. 22nd IEEE international conference on tools with artificial intelligence, Arras, France. 2010.

31. Smets P. Constructing the pignistic probability function in a context of uncertainty. Univ Libre Brux. 1989;89:29-40.

32. Mercier D, Cron G, Denoeux T, et al. Fusion of multi-level decision systems using the transferable belief model. 7th International conference on information fusion, Philadelphia, PA, USA. 2005.

33. Simon C, Weber P. Evidential networks for reliability analysis and performance evaluation of systems with imprecise knowledge. IEEE Trans Reliab. 2009;58:69-87.

34. Voorbraak F. A computationally efficient approximation of Dempster-Shafer theory. Int J Man Mach Stud. 1989;30:525-36.

35. Appriou A. Uncertainly Theories and multi-sensor data fusion. (1stedn), John Wiley& Sons, New York, USA. 2014.

36. Zlatoff N, Tellez B, Baskurt A. Combining local belief from low-level primitives for perceptual grouping. Pattern Recognit. 2008;41:1215-29.