Deep Learning for Infrared Thermal Image Based Machine Health Monitoring

Olivier Janssens, Mia Loccufier, Rik Van de Walle and Sofie Van Hoecke

***Abstract*—The condition of a machine can automatically be identified by creating and classifying features that summarize characteristics of measured signals. Currently, experts, in their respective fields, devise these features based on their knowledge. Hence, the performance and usefulness depends on the expert's knowledge of the underlying physics, or statistics. Furthermore, if new and additional conditions should be detectable, experts have to implement new feature extraction methods. To mitigate the drawbacks of feature engineering, a method from the sub-field of feature learning, i.e. deep learning, more specifically convolutional neural networks, is researched in this article. The objective of this article is to investigate if and how deep learning can be applied to infrared thermal video to automatically determine the condition of the machine. By applying this method on infrared thermal data in two use cases, i.e. machine fault detection and oil level prediction, we show that the proposed system is able to detect many conditions in rotating machinery very accurately (i.e. 95 % and 91.67 % accuracy for the respective use cases) without requiring any detailed knowledge about the underlying physics, and thus having the potential to significantly simplify condition monitoring using complex sensor data. Furthermore, we show that by using the trained neural networks, important regions in the infrared thermal images can be identified related to specific conditions which can potentially lead to new physical insights.**

I. Introduction

CONDITION MONITORING (CM) of a machine and its components is crucial to avoid downtime and unnecessary costs, enhance the machine's lifetime and improve safety by recognizing abnormal behaviour of a machine or machine component. This generally implies the comparison of healthy and faulty situations indicated by processed measurements, either manually obtained through operators with portable devices or continuously through built-in sensors. From these measurements, informative characteristics (features) are extracted (engineered) by a CM expert that have to be interpreted to determine the machine's condition. To automate condition monitoring, the streams of measurements need to be processed automatically, requiring a system that automatically extracts features from the streams of measurements and provide these to a machine learning algorithm that determines the machine's condition. Before such an algorithm can be used, it has to be trained in order to be able to detect the different conditions: (1) first, a data set of measurements with accompanying labels is created that indicate the different machine conditions; (2) the algorithm is subsequently trained, using extracted features from the data set, to detect the different conditions using streams of measurements. A block diagram illustrating this process is given in Figure 1. As can be seen in this figure, sensory data (X) is gathered together with the corresponding labels (Y), i.e. the machine's condition during the measurements. From the data, features are extracted (ϕ) which are subsequently given to a machine learning algorithm together with the labels (Y). The machine learning algorithms will learn a model (θ) that can distinguish between the different conditions. When the system is put into production, new measurements will be taken for which the conditions are unknown. In order to detect the condition, features are extracted out of the measurements and fed to the model that is now capable of predicting the condition of the machine ($\hat{Y}$).

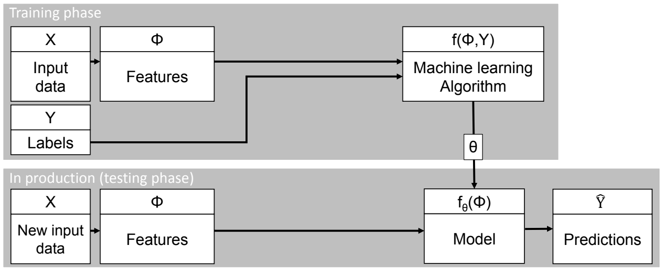


Fig. 1: Block diagram illustrating feature engineering and the training of a machine learning model for condition monitoring.

Currently, the features extracted from the measurements are engineered by condition monitoring experts. The feature engineering process can either be data-driven or model-driven. Data-driven feature engineering entails creating features that describe measurable characteristics that are related to a condition by observation without taking the underlying physics into account. Model-driven feature-engineering requires reasoning on the underlying physics of the conditions to deduce what resulting phenomena can occur and how to quantify them in features.

A lot of work and research [1], [2] has already been done to understand the underlying physics for vibration analysis, a commonly used technique for CM, resulting in many useful signal processing techniques and methodologies for model-driven feature engineering. But also data-driven feature engineering is already successfully applied [3], [4] on vibration data by, for example, extracting well known statistics from the vibration signals (e.g. mean, standard deviation, median and skew).

As certain faults in rotating machinery remain difficult to detect using vibration signals alone, for example lubrication

O. Janssens, R. Van de Walle and S. Van Hoecke are with imec - Ghent University - IDLab, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium, e-mail: (odjansse.janssens@ugent.be)

Mia Loccufier is with the DySC Research Group, Ghent University, Technologiepark 914, 9052 Zwijnaarde, Belgium

related problems [5], other types of sensors, like thermal imaging, can be considered. Interpreting the phenomena in the infrared thermal (IRT) images requires however substantial insights into the mechanics and thermodynamics of the systems. As complete physics-based modelling of a machine is difficult and requires a lot of knowledge, effort and time [6], mainly data-driven feature engineering has been applied for IRT-based condition monitoring [7], [8], [9], [5].

Despite the successful applications of feature engineering, both in vibration and IRT data, there is a possibility that a feature-engineering-based system will not perform optimally for two reasons: (1) the feature engineering depends on an expert with knowledge about mechanical engineering or statistics who might not be able to devise features that fully describe the dynamics of the signals that are required for correct classification; (2) it is possible that the required knowledge to create features is not available yet. Furthermore, a CM system can be designed for a specific set of conditions, hence, when new conditions should be detectable an expert has to implement new feature extraction capabilities into the system.

In order to circumvent this problem, feature learning/representation learning can be employed. As opposed to feature engineering, wherein a human creates the features, feature learning uses a machine learning algorithm to learn and create useful features from raw data. The algorithm learns features without human input that optimally represent the raw data for the required task and has therefore the potential to be more powerful than manually engineered features. It should be noted that feature learning is different from feature selection that is used to select the most informative subset of features from all the available (manually engineered) features; so there is no feature learning during feature selection. A schematic representation, illustrating the difference between feature engineering without, resp. with, feature selection, and feature learning, can be seen in Figure 2. Feature engineering takes input data (X) and extracts features (ϕ) that are subsequently used to train a classifier ($f_\theta(.)$) with learnable parameters (θ) that outputs predictions ($\hat{Y}$). When feature selection is applied, the most informative subset of features is selected ($\psi \subseteq \phi$) to train the classification algorithm. Conversely, feature learning will not extract features, but will use raw input data and transform it using $t_{\theta_1}(.)$ wherein θ_1 consists of the learnable parameters of the transformation. The transformation provides a new representation of the input data, better suited for the classification task. The transformation steps can be repeated many times—each with their own set of learnable parameters— in order to transform the data optimally during classification, i.e., optimal features are learned for the classification task.

In recent years feature learning has become very popular by the introduction of deep learning [10] (DL). DL methods are representation-learning methods with multiple levels of representation, obtained by composing simple non-linear modules that each transform the representation (of the data) at one level (starting with the raw input) into a representation at a higher, slightly more abstract level. By composing multiple transformations, very complex functions can be learned for handling complex data such as images and video. DL is done using various types of deep neural networks (DNN) wherein every layer will learn a new representation of the data (i.e. learn features). DL achieves good results in image recognition, speech recognition, drug discovery, analysing particle accelerator data and natural language processing [10]. All these applications require complex data with many variables, resulting in the success of deep learning. In previous work we have applied techniques of deep learning, i.e. a convolutional neural network, on vibration signals to detect machine faults [11]. As vibration signals are one dimensional signals, a shallow convolutional network sufficed. IRT data, however, has many variables (i.e. pixels), and therefore form perfect fit for researching convolutional neural networks as a tool for condition monitoring. A vast amount of data is required to train a very deep neural network. To mitigate this problem, we investigate transfer learning, a solution when needing a lot of data, and its usability for IRT data. We apply our proposed DNN on two use cases, namely machine fault detection and oil level prediction, and show that by using feature learning, and thus not manually designing features but letting a DNN learn the features, a significant condition detection performance gain can be achieved. Finally, we show that the NNs actually focus on certain parts in the IRT images to make the classification decisions, which can potentially lead to new physical insights.

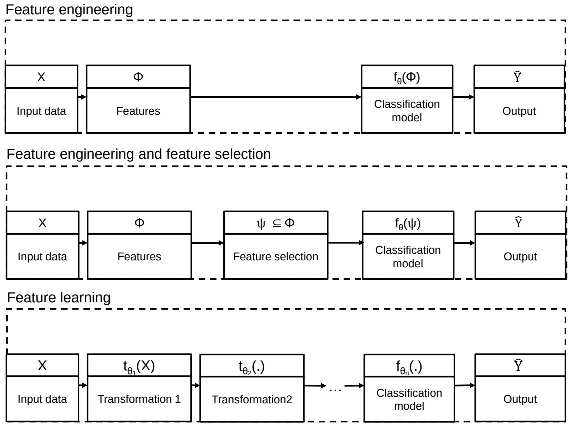


Fig. 2: Schematic representation of feature engineering without, resp. with feature selection, and feature learning.

The remainder of the paper is as follows. Section II discusses an overview of the recent progress regarding neural networks for condition monitoring. In Section III, our DNN is discussed together with the transfer learning approach, and insights into the decision making process are visualised. The two subsequent sections (Section IV and V) present the two use cases on which the DNNs are applied. Finally, in Section VI, a conclusion is provided.

II. Neural Networks for Condition Monitoring

Neural networks (NN) have been used for many decades. However, most often they are used in combination with features engineered by an expert [12], [13]. In contrast, feature learning uses a raw representation of the input data and lets an algorithm learn and create a suitable representation of the data, i.e. features. An example of such a process using NNs is given in [14], wherein vibration spectrum images are created and given to NNs for rolling element bearing (REB) fault classification. Feature learning can be done by using both supervised and/or unsupervised methods. For REB fault detection using vibration measurements, unsupervised methods using auto-encoders have been used recently [15]. Auto-encoders are NNs which are designed to replicate the given input. The NN has a single hidden layer containing less nodes than the input layer. The purpose of this hidden layer is to learn a compressed representation of the input data. An auto-encoder is used to extract features which are given to a classification algorithm. It should be noted that many auto-encoders can be stacked on top of each-other to form a DNN. Each layer is trained individually, as training an entire DNN at once suffers from the gradient vanishing problem. During NN training, the (local) minimum of the error function is found by iteratively taking small steps (i.e. gradient descent) in the direction of the negative error derivative with respect to the network's weights (i.e. gradients). To calculate this gradient, back-propagation is used. Back-propagation in essence is the chain-rule applied to a NN. Hence, the gradient is propagated backward through each layer. With each subsequent layer, the magnitude of the gradients get exponentially smaller (vanishes), making the steps also exponentially smaller, resulting in very slow learning of the weights in the lower (first) layers of a DNN. An important factor causing the gradients to shrink are the activation function derivatives (i.e. derivative of a layer's output with respect to its input). When the sigmoid activation function is used in the network, the magnitude of the sigmoid derivative is well below 1 in the function's range causing the gradient to vanish. To solve this problem, in 2012 Krizhevsky et al. [16] proposed another type of activation function, called the rectified linear unit which does not suffer from this problem. Hence, the vanishing gradient problem was mostly solved, enabling much deeper (supervised) NNs to be trained as a whole, resulting in many new state-of-the-art results.

A neural network is commonly dense and fully connected, meaning that every neuron of a layer is connected to every other neuron in the subsequent layer. Each connection is a weight totalling many parameters. This number of parameters is difficult to train as the network will memorize the data (overfitting), especially when too little data is available. If possible, a partial solution to this problem is to gather more data. Nevertheless, the training procedure will take very long. Another partial solution is implicitly provided in convolutional neural networks (CNN) [17]. CNNs are designed to deal with images and therefore exploit certain properties, i.e. local connectivity, weight sharing, and pooling, which results in a faster training phase, but also less parameters to train:

- **Local connectivity**: When providing an image as input, instead of connecting every neuron in the first hidden layer to every pixel, a neuron is connected to a specific local region of pixels called a local receptive field. A local receptive field has a grid structure with a height (h), width (w) and depth (d) and is connected to a hidden neuron in the next layer. Such a local receptive field is slid across the input grid structure (i.e. image). Each local receptive field is connected to a different hidden neuron, where each connection is again a weight.
- **Weight sharing**: Weights (also called kernel or filter) consist of a grid structure equal to the size of a local receptive field. Instead of having a unique set of weights for each location in the input grid structure, the weights are shared. As any other image processing filter, weights in a CNN will extract features from the input. Due to weight sharing, the same feature can be extracted in different locations of the input. The output of such a transformation is called a feature map. It should be noted that in a CNN, every layer will have multiple sets of weights so that a multitude of features can be extracted resulting in multiple feature maps (k). Due to weight sharing the amount of weights in the NNs are reduced.
- **Pooling**: Pooling is done after a convolutional layer and reduces the dimension of the feature maps. It is applied by sliding a small window over the feature maps while extracting a single value from that region by the use of for example a max or mean operation. A feature map will hence reduce in size resulting in less parameters and reduced number of computations in subsequent layers.

For more information on CNNs, we refer the reader to [17].

Data sets are often very small for tasks in specialized fields compared to the required amount of data to train a DNN. Hence, DNNs will tend to overfit. To overcome this problem, pre-trained networks can be used which are NNs trained for another task for which a lot of data was available. In essence, the weights of the already trained network are re-purposed for the new tasks. It has been shown that such a NN will have learned general features that can be used for other tasks [18], [19]. It has also been shown that NNs, which are trained on images of everyday scenery, can be re-purposed and modified to be applicable in tasks which require domain specific images, such as medical images [20] or aerial images [21]. The process of re-using and modifying a trained NN is called transfer learning. There are several methods to apply transfer learning [18]:

- Remove the last layer (k) or multiple layers (k-t, ...,k). Hence, by providing the modified pretrained NN with input samples the network will output intermediary abstract representations of the data that can be given to a new classifier such as a support vector machine. The idea behind this approach is that the network has learned re-usable features, which at a certain layer are useful for the task at hand, and that only a new classifier has to be trained using the re-usable features.
- In addition to removing one or more layers, it is also possible to attach new layers to the modified pre-trained network. The idea behind this method is that the initial layers have learned useful weights, but that the subsequent layers have not. Hence, they have to be replaced and trained.
- Following on the method above, one can choose to only

train the newly added layers (using gradient descent in combination with back-propagation) in order to modify the weights of these new layers without modifying the weights of the transferred layers.

- As opposed to only training the newly added layers, it is also possible to train the entire network, i.e. train the pre-trained layers and new layers. The idea behind this method is that neighbouring layers co-adapt during training, which can only be done when training all layers [18].

The application of transfer learning in this paper is discussed in the next section.

III. Neural Network Architecture

Images are complex data as they consist of many variables (pixels), hence a deep network is required. However, we determined that the data sets we constructed in the two use cases contained too little data to properly train a DNN for the IRT data. Gathering enough data is infeasible. Hence, research into transfer learning for IRT is done.

Various transfer learning methods were tested, however, the last option discussed in Section II, i.e. training both the pre-trained and new layers, provided the best results. We opted to use a pre-trained VGG (neural network created by the Visual Geometry Group at University of Oxford) [22] network which achieves state-of-the-art results on the imagenet data set. The VGG network is a very deep CNN containing 16 layers, that was trained on natural images. The goal of the VGG network was to classify images in one of a thousand categories. The VGG network uses rectified linear activation functions in every layer except the last layer, which is a fully-connected layer where softmax activation functions are used. A layer with softmax activation functions provides a probabilistic mutually exclusive classification, i.e. it provides 1000 values ranging between 0 and 1 and the sum of these thousand values is equal to one. Hence, it gives the probability of a sample belonging to a certain class.

For transfer learning purposes, the last layer of the VGG network was removed as our data-set has fewer classes. A new fully-connected layer was attached to network. This new layer also uses softmax activation functions, but less weights, as there are less classes to distinguish for the task at hand. In the end, this means that all except for one layer of the VGG network (which are pre-trained) are reused in our network and solely the last layer is new. In Figure 3, the architecture of the network can be seen. As has been demonstrated in other research, the fact that a network's layers have been trained using a certain type of images, does not mean that transfer learning is not possible for totally different types of images. Hence, we hypothesize that a pre-trained DNN, such as the VGG network, can be re-used for machine condition detection using IRT images.

As the input layer of the VGG network is re-used, our dataset has to be preprocessed according to the data that was initially provided to the original VGG network, hence preprocessing as described in [22] was applied. Images (i.e. frames) are pre-processed by removing the mean value. Next smoothing is applied using a Gaussian kernel with a standard deviation of 3 pixels. Then all frames are aligned to a common reference frame (i.e. image registration) and subsequently cropped to a width and height of 224 pixels.

Training is applied using mini-batch gradient descent, updating all the weights of the network, including the weights of the pre-trained layers. However, the learning rate for the mini-batch gradient descent algorithm should be smaller than the original learning rate to minimally influence the already pre-trained layers. Therefore, it was set to 1.10^{-5}. The network was trained using a mini-batch size of 8 and for 100 epochs.

A. Insights into Infrared Thermal Data

It is difficult to know where to look on an infrared thermal image in order to detect a specific machine condition. NNs are nevertheless able to discover what is important in the images to make a decision regarding the conditions. Thus it can be concluded that the necessary information is present in the thermal images. Extracting the regions in an image that are important for a NN, by applying the technique proposed by Zeiler et al. [23], can potentially lead to new physical insights. The Zeiler method has three steps that are iterated over:

- The first step masks a part of the input image (i.e. a 7×7 square of pixels is set to a constant value).
- In step two, the modified incomplete image is classified by the trained CNN. The CNN has softmax activation functions in the output layer which give a probability for every possible class.
- In the third step the class probability corresponding to the correct class is saved in a matrix with the same dimensions as the image. The probabilities are stored in the location corresponding to the location that was masked in the original image.

These three steps are iterated over so that every part of the image is masked once. The idea behind this method is that if an important and crucial part of the image is masked, the probability for the correct class will be low (i.e. closer to zero). Hence, if such a drop in probability is observed when a specific part of the image is masked, it can be concluded that said part of the image is crucial for that particular class. An intuitive example is given in [23], where a CNN is trained to detect objects in natural images. One of the possible classes is "dog". Hence, if a picture of a dog is given to the NN where the face of the dog is hidden by the mask, the probability for the class "dog", provided by the network, will be much lower compared to the case when the dog's face is not masked.

In the next section the first use case, wherein the CNN presented above, is discussed.

IV. Use case one: Machine fault detection

In this use case, IRT video is recorded for various conditions in a rotating machinery set-up. The CNN uses the IRT data to detect the condition of a rolling element bearing and the gradation of imbalance in the machine. Two separate data sets were created using the same set-up but on separate moments in time. The conditions present in each data set are listed in Table I and Table II respectively.

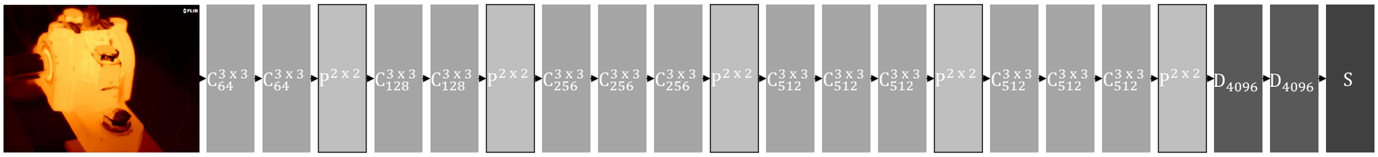


Fig. 3: Architecture of the deep convolutional neural network for IRT CM. $C_k^{h\times w}$ denotes a convolutional layer with k feature maps and receptive field of dimension $h \times w$. P denotes a pooling layer. D_n denotes a dense fully connected layer with n neurons. S denotes a softmax layer.

TABLE I: Summary of the 8 conditions in data set one

	No imbalance	**Imbalance: 13 g or 17.3 N**
Healthy REB (HB)	Condition 1	Condition 2
Outer-raceway fault (ORF)	Condition 3	Condition 4
Mildly inadequately lubricated bearing	Condition 5	Condition 6
Extremely inadequately lubricated bearing	Condition 7	Condition 8

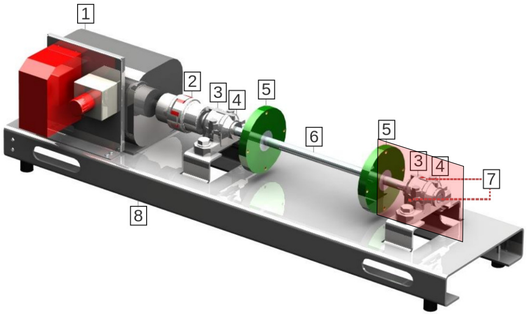


Fig. 4: 3D image of the set-up. The labels are: 1. servo-motor; 2. coupling; 3. bearing housing; 4. bearing; 5. disk; 6. shaft; 7. thermocouple; 8. metal plate. The red square indicates what the IRT camera records

A. *Set-up*

The set-up can be seen in Figure 4, for which the rotation speed was set to 25 Hz. The REB in the housing at the right-hand side in the set-up is changed in-between test runs, hence this is the housing that is monitored by the thermal camera. Additional to the IRT camera, two thermocouples are mounted to measure the ambient temperature.

The type of bearings used were spherical roller bearings and to imitate outer-raceway faults (ORF) in the REBs, three small shallow grooves were added mechanically on the REBs' outer-raceway (see Figure 5 for an example of such a groove). The ORF is placed at the 10 o'clock position in the housing (i.e. close to the top of the housing, facing the IRT camera) for data set one and at the 6 o'clock position (i.e. loaded zone) for data set two. Lubricant grease is added to every REB. The required amount of grease is 2.5 g as is discussed in [5].

Both the healthy bearings (HB) and those with an ORF are placed in a housing which contains a grease reservoir. The grease reservoir contains 20 g [24]. For the REBs with reduced lubricant in data set one, i.e. mildly inadequately lubricated bearing (MILB) and extremely inadequately lubricated bearing (EILB), no grease reservoir is present. For the MILBs the grease on each individual REB is superficially removed (1.5 g reduction). Similarly, for the EILBs the grease in the REBs is decreased more (0.75 g reduction). For the hard particle faults in data set two, 0.02 g of iron particles are mixed in the lubricant of the REBs.

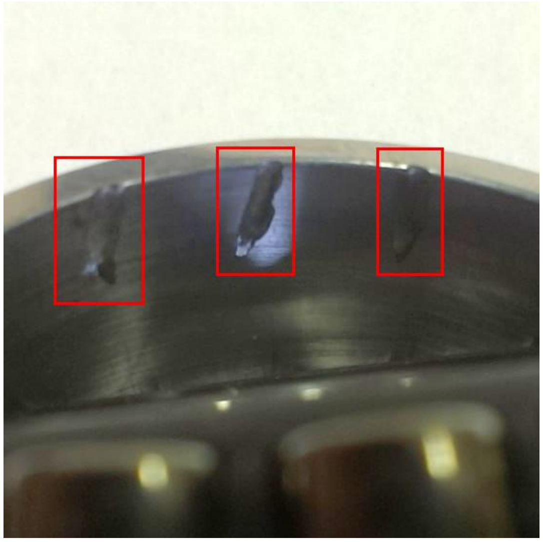


Fig. 5: Three shallow grooves in the outer-raceway of a bearing simulating an outer-raceway fault.

To complete the data sets, all the different REB conditions are also tested during imbalance, this is done by adding bolts to the rotor at a radius of 5.4 cm. The weight of the bolts can be seen in Table I and II.

B. *Data set*

To construct data sets that are large enough to validate the proposed methods, every condition in both data sets are created for five different REBs. By using multiple REBs, variability is introduced in the data set due to manufacturing, mounting and grease distribution. Each REB is run for one hour, and the last 10 minutes —when steady-state is reached— is captured by the IRT camera. For data set one, in total, 5 REBs × 8 conditions = 40 IRT recordings are made. For data set two, 5 REBs × 12 = 60 recordings are made. It should also be noted that relative temperatures are used and not absolute temperature. This is done by subtracting the temperature measured by the thermocouple from the temperatures measured by the thermal camera. For more information regarding the data set one we refer the reader to [5]. It should be noted that simultaneously with the IRT measurements, accelerometer measurements were captured to check the validity of the data set.

TABLE II: Summary of the 12 conditions in data set two

	No imbalance	**Imbalance: 4.1 g or 5.5 N**	**Imbalance: 9.3 g or 12.4 N**	**Imbalance: 13 g or 17.3 N**
Healthy REB (HB)	Condition 1	Condition 2	Condition 3	Condition 4
Outer-raceway fault (ORF)	Condition 5	Condition 6	Condition 7	Condition 8
Hard particle contamination (HP)	Condition 9	Condition 10	Condition 11	Condition 12

C. Application of the convolution neural network

To detect the multitude of conditions in the two data sets, two CNNs are required. One CNN, as described in Section III, is trained to detect the different REB conditions. A second CNN's goal is to detect the gradation of imbalance and is trained using differenced frames. It was determined that imbalance could not be detected by solely using spatial information of the heat distribution in the component. Temporal information is required to make the vibrations, due to imbalance, visible in the images. Hence, subsequent images are differenced (i.e. $I_{t-1} - I_t$). By differencing these frames, the movement in the images becomes visible as described in [5].

By combining the outputs of the two CNNs (i.e. one focussing on the spatial aspects and one focussing on the temporal aspects) an overall classification can be done.

D. Results

To put the results of the CNN approach in perspective, they are compared to a feature engineering approaches given in [5]. Accuracy (see Equation 1) is used as metric for fault detection performance. This metric specifies the ratio between the number of samples that are correctly classified and all the samples in total. The scores were determined during five-fold cross-validation. This means that the CNNs were trained on recorded data from REB two, three, four and five and subsequently the CNNs were tested on data from REB one. This is done five times so that every bearing is in the test-set once. For more information on this evaluation procedure we refer the reader to [5].

$$\text{accuracy=}\frac{\text{Number of correctly classified frames}}{\text{Total number of frames that were classified}} \quad (1)$$

The results for both the feature engineering based approach and the feature learning based approach on data set one are listed in Table III. The detection of imbalance can be done perfectly (accuracy 100 %) using either feature engineering or feature learning. However, for the detection of the REB condition, feature learning achieves better results (7 % higher accuracy). Overall, for all 8 conditions together, the feature learning approach thus provides a 7 % better result.

Results for both the feature engineering based approach and the feature learning based approach on data set two are listed in Table IV. It can be seen that feature learning provides way better results for both the detection of the imbalance gradation and the detection of the specific REB condition. In the end, the feature learning approach provides a 37 % better accuracy compared to the feature engineering approach.

In general it can be concluded that the CNN approach gives very good results on both data sets without requiring expert knowledge about the problem. However, as a downside, NNs are black box systems, meaning that their inner-workings are not human-interpretable. Nevertheless, insights can be derived from NNs using the method described in Section III-A.

TABLE III: Results of both the feature engineering (FE) and feature learning (FL) based approach on data set one. σ denotes the standard deviation.

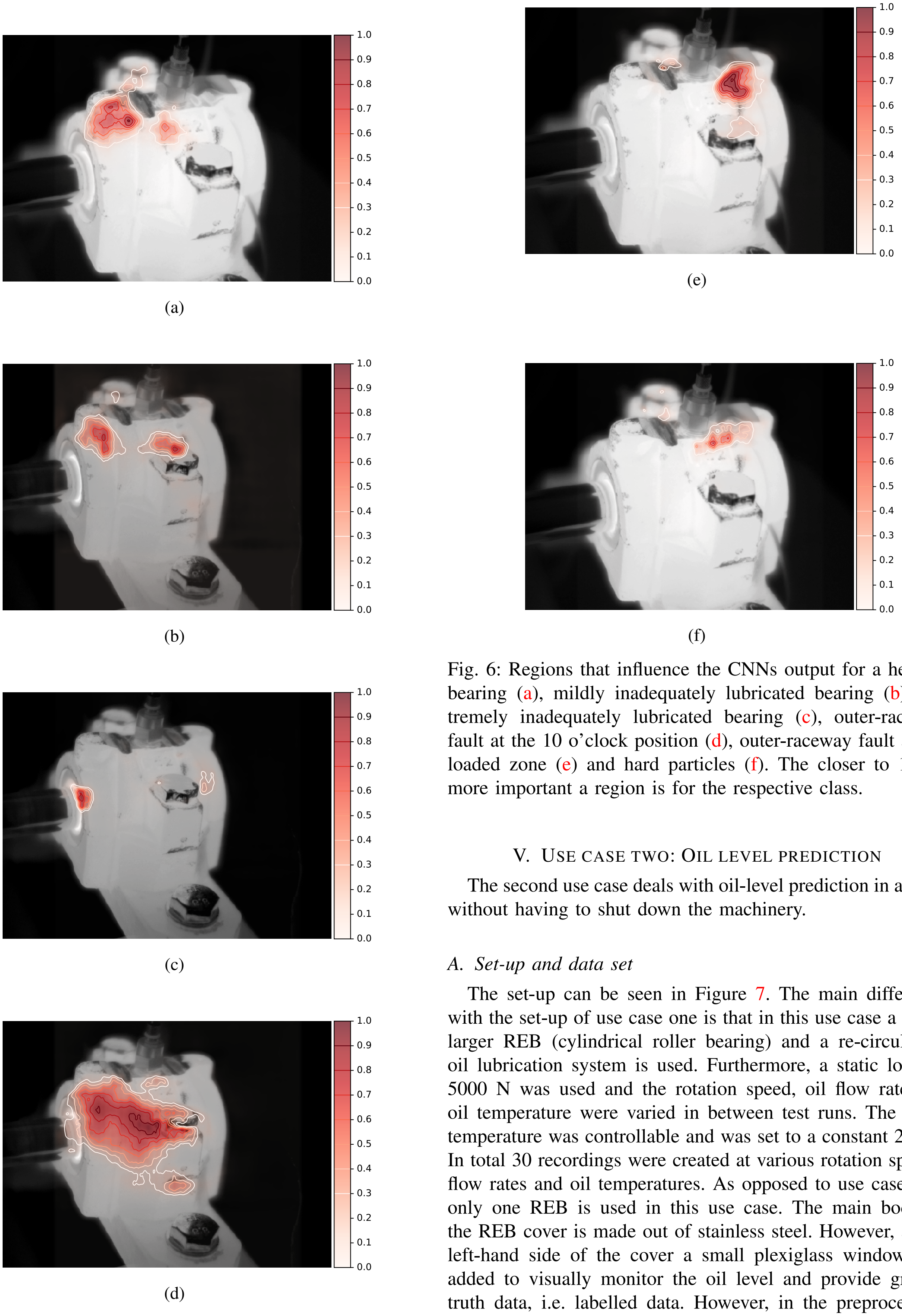
Method	**Conditions**	**Accuracy**
FE	MILB, EILB, HB, ORF	88.25 % ($\sigma = 8.07$ %)
FL	MILB, EILB, HB, ORF	95.00 % ($\sigma = 6.12$ %)
FE	balance and imbalance	100.0 % ($\sigma = 0.00$ %)
FL	balance and imbalance	100.0 % ($\sigma = 0.00$ %)
FE	All 8 conditions	88.25 % ($\sigma = 8.07$ %)
FL	All 8 conditions	95.00 % ($\sigma = 6.12$ %)

TABLE IV: Results of both the feature engineering (FE) and feature learning (FL) based approach on data set two. σ denotes the standard deviation.

Method	**Conditions**	**Accuracy**
FE	HP, ORF, HB	65.00 % ($\sigma = 16.16$ %)
FL	HP, ORF, HB	98.33 % ($\sigma = 3.33$ %)
FE	Imbalance gradation	88.33 % ($\sigma = 12.47$ %)
FL	Imbalance gradation	93.33 % ($\sigma = 9.72$ %)
FE	All 12 conditions	55.00 % ($\sigma = 11.31$ %)
FL	All 12 conditions	91.67 % ($\sigma = 9.13$ %)

In Figure 6 the output based on this method is visualized for the six bearing conditions. The figures indicate which parts are important in the IRT image for the specific conditions. For example, to identify if a REB is extremely inadequately lubricated, the area around the seal is very important (Figure 6c), which can for example be due to the heat originating from the increased friction between the shaft and the seal. Another example is the large area for an outer-raceway fault at the 10 o'clock position (Figure 6d). Due to the fact that the ORF is actually facing the camera inside the housing, a possible increase in heat is observable in this area. In general, these locations can help to make a link to the underlying physics and can potentially lead to new insights. However, further research is needed to relate each highlighted image part with the specific underlying physical phenomenon.

When testing our method using a Nvidia GeForce GTX TITAN X, 122.26 frames per second can be processed with a standard deviation of 7.27 frames per second, showing that the presented method can be used for real-time condition monitoring.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 6: Regions that influence the CNNs output for a healthy bearing (a), mildly inadequately lubricated bearing (b), extremely inadequately lubricated bearing (c), outer-raceway fault at the 10 o'clock position (d), outer-raceway fault at the loaded zone (e) and hard particles (f). The closer to 1, the more important a region is for the respective class.

V. Use case two: Oil level prediction

The second use case deals with oil-level prediction in a REB without having to shut down the machinery.

A. *Set-up and data set*

The set-up can be seen in Figure 7. The main difference with the set-up of use case one is that in this use case a much larger REB (cylindrical roller bearing) and a re-circulatory oil lubrication system is used. Furthermore, a static load of 5000 N was used and the rotation speed, oil flow rate and oil temperature were varied in between test runs. The room temperature was controllable and was set to a constant 23 °C. In total 30 recordings were created at various rotation speeds, flow rates and oil temperatures. As opposed to use case one, only one REB is used in this use case. The main body of the REB cover is made out of stainless steel. However, at the left-hand side of the cover a small plexiglass window was added to visually monitor the oil level and provide ground truth data, i.e. labelled data. However, in the preprocessing phase the plexiglass part is removed from the IRT image. For

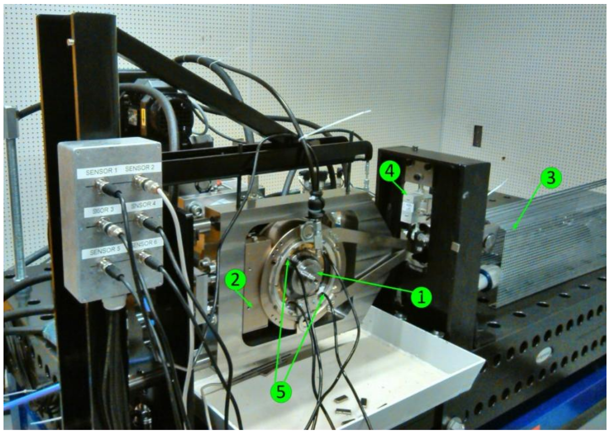


Fig. 7: Image of the used set-up. 1. bearing, 2. hydrostatic pad to apply radial load on the bearing, 3. pneumatic muscle for loading the bearing, 4. force cell for friction torque, 5. temperature measurements.

more information on the set-up and dataset we refer the reader to [25]. The goal is to let the CNN, described in Section II, automatically determine if the oil-level in the REB is full or not as this is not determinable visually by humans. The same training and preprocessing procedures as described for use case one are applied.

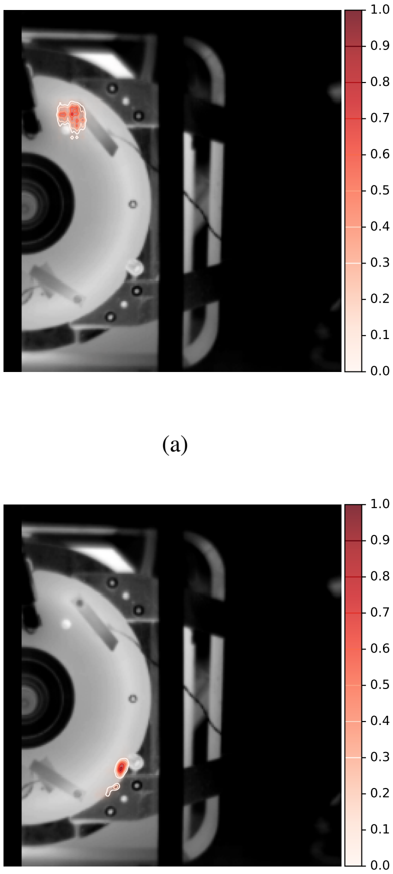
B. Results

The accuracy score was determined using leave-one-out cross-validation as the variability in conditions of the data-set are rather large for the amount of samples. To put the feature learning results in perspective also a feature engineering based approach is used similar to the one discussed in [5] where general statistical features are used. The results can be seen in Table V. As can be seen a feature learning based approach provides better results (6.67%). Not only does feature learning provide better results, it also does not require an expert to engineer features.

TABLE V: Results of both the feature engineering (FE) and feature learning (FL) based approach in use case two.

Method	Conditions	Accuracy
FE	Full, not-full	80.00 %
FL	Full, not-full	86.67%

The search for important parts in the image responsible for the classification result resulted in the Figure 8a and 8b for a REB full of oil and a REB not full of oil. In contrast to use case one, the underlying physics of these images can be interpreted. To detect if a REB is full of oil, the top side of the REB is important, which is to be expected as this part will be hotter when the REB is full of oil. Conversely, to detect if the REB is not full, the bottom part of the REB is important as only this part will be significantly warmer when the REB is not full of oil.



(a)

(b)

Fig. 8: Regions that influence the CNNs output for a REB full of oil (a) and not full of oil (b)

VI. Conclusion

In this article it is shown that convolutional neural networks, a feature learning tool, can be used to detect various machine conditions. The advantage of feature learning is that no feature engineering or thus expert knowledge is required. Feature engineering can also result in a sub-optimal system especially when the data is very complex, such as for thermal infrared imaging data.

Deep neural networks, such as convolutional neural networks, require a vast amount of data to train. To mitigate this problem we investigated transfer learning, which is a method to re-use layers of a pre-trained deep neural network. We show that by using transfer learning, wherein layers of a trained CNN on natural images are re-purposed, the convolutional neural network outperforms classical feature engineering in both the machine fault detection and the oil level prediction use case. For both use cases, the feature learning approach provides at least a 6.67% better accuracy compared to the feature engineering approach, and even up to 37% accuracy improvement for dataset two of use case one.

Finally, as it is difficult to know where in the image to look at to detect certain condition, we show that by applying the method of Zeiler et al. [23] on the trained convolutional neural networks valuable insights into the important regions of the thermal images can be detected, potentially leading to new physical insights.

The presented method has the potential to improve online condition monitoring in for example offshore wind turbines. The maintenance costs for offshore wind turbines is very high due to the limited accessibility. Installing an infrared thermal camera in the offshore wind turbine's nacelle, combined with the presented method, allows for online condition monitoring. Another potential application is the monitoring of bearings in manufacturing lines. Using thermal imaging together with the method of Zeiler et al. applied to the trained convolutional neural network allows identifying the location of the faults in the manufacturing lines.

Acknowledgment

This work was funded by the VIS O&M Excellence project of Flanders Innovation & Entrepreneurship (VLAIO), and has been performed in the framework of the Offshore Wind Infrastructure Application Lab (http://www.owi-lab.be).

References

[1] W. A. Smith and R. B. Randall, "Rolling element bearing diagnostics using the case western reserve university data: A benchmark study," *Mechanical Systems and Signal Processing*, vol. 64, pp. 100–131, 2015.

[2] E.-T. Idriss and J. Erkki, "A summary of fault modelling and predictive health monitoring of rolling element bearings," *Mechanical Systems and Signal Processing*, vol. 6061, pp. 252 – 272, 2015.

[3] R. Heng and M. Nor, "Statistical analysis of sound and vibration signals for monitoring rolling element bearing condition," *Applied Acoustics*, vol. 53, no. 1–3, pp. 211 – 226, 1998.

[4] Y. L. Murphey, M. A. Masrur, Z. Chen, and B. Zhang, "Model-based fault diagnosis in electric drives using machine learning," *IEEE/ASME Transactions On Mechatronics*, vol. 11, no. 3, pp. 290–303, 2006.

[5] O. Janssens, R. Schulz, V. Slavkovikj, K. Stockman, M. Loccufier, R. V. de Walle, and S. V. Hoecke, "Thermal image based fault diagnosis for rotating machinery," *Infrared Physics & Technology*, vol. 73, pp. 78 – 87, 2015.

[6] W. Moussa, "Thermography-assisted bearing condition monitoring," Ph.D. dissertation, Université d'Ottawa/University of Ottawa, 2014.

[7] A. Widodo, D. Satrijo, T. Prahasto, G.-M. Lim, and B.-K. Choi, "Confirmation of Thermal Images and Vibration Signals for Intelligent Machine Fault Diagnostics," *International Journal of Rotating Machinery*, vol. 2012, pp. 1–10, 2012.

[8] V. T. Tran, B.-S. Yang, F. Gu, and A. Ball, "Thermal image enhancement using bi-dimensional empirical mode decomposition in combination with relevance vector machine for rotating machinery fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 38, no. 2, pp. 601–614, Jul. 2013.

[9] G.-m. Lim, Y. Ali, and B.-s. Yang, "The Fault Diagnosis and Monitoring of Rotating Machines by Thermography," J. Mathew, L. Ma, A. Tan, M. Weijnen, and J. Lee, Eds. Springer London, 2012, pp. 557–565.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] O. Janssens, V. Slavkovikj, B. Vervisch, K. Stockman, M. Loccufier, S. Verstockt, R. V. de Walle, and S. V. Hoecke, "Convolutional neural network based fault detection for rotating machinery," *Journal of Sound and Vibration*, vol. 377, pp. 331 – 345, 2016.

[12] B. Li, M.-Y. Chow, Y. Tipsuwan, and J. C. Hung, "Neural-network-based motor rolling bearing fault diagnosis," *IEEE transactions on industrial electronics*, vol. 47, no. 5, pp. 1060–1069, 2000.

[13] Z. Chen, C. Li, and R.-V. Sanchez, "Gearbox fault identification and classification with convolutional neural networks," *Shock and Vibration*, vol. 2015, p. 10, 2015.

[14] M. Amar, I. Gondal, and C. Wilson, "Vibration spectrum imaging: A novel bearing fault classification approach," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 494–502, 2015.

[15] N. Verma, V. Gupta, M. Sharma, and R. Sevakula, "Intelligent condition based monitoring of rotating machines using sparse auto-encoders," in *IEEE Conference on Prognostics and Health Management (PHM)*, 2013, pp. 1 – 7.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097 – 1105.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[19] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.

[20] W. Zhang, R. Li, T. Zeng, Q. Sun, S. Kumar, J. Ye, and S. Ji, "Deep model based transfer and multi-task learning for biological image analysis," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 1475–1484.

[21] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14 680–14 707, 2015.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[23] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision ECCV 2014*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 818–833.

[24] Schaeffler, "Fag split plummer block housings of series snv," Online, 2015.

[25] O. Janssens, M. Rennuy, S. Devos, M. Loccufier, R. Van de Walle, and S. Van Hoecke, "Towards intelligent lubrication control: Infrared thermal imaging for oil level prediction in bearings," in *Proceedings of the IEEE Multi-Conference on Systems and Control*. IEEE, 2016.

Olivier Janssens received his Master degree in industrial engineering focussing on information and communication technology from the University College of West Flanders in 2012. Following his studies he joined the IDLab at department of Electronics and Information Systems (ELIS), Ghent University-imec, in order to research multi-sensor data-driven condition monitoring methods.

Mia Loccufier received a M.S degree of electromechanical engineer, a M.S. degree of automatic control engineer and a PhD degree in electromechanical engineering from Ghent University . She is a professor at the DySC research group of the Department of Electrical Energy, Systems and Automation, Faculty of Engineering, Ghent University, Belgium. She is a lecturer in mechanical vibrations, structural dynamics and systems dynamics. The field of research is focussed on the dynamics of technical systems. The main research themes are passive control, especially nonlinear tuned mass dampers of mechanical systems and structures , dynamics of rotating machinery ; stability and bifurcation analysis of nonlinear systems and structures, control of underactuated mechanical systems.

Rik Van de Walle received his M.Sc. and PhD degrees in Engineering from Ghent University, Belgium in 1994 and 1998 respectively. After a visiting scholarship at the University of Arizona (Tucson, USA), he returned to Ghent University, where he became professor of multimedia systems and applications, and head of the Multimedia Lab. His current research interests include multimedia content delivery, presentation and archiving, coding and description of multimedia data, content adaptation, and interactive (mobile) multimedia applications.

Sofie Van Hoecke received her Master degree in Computer Science from Ghent University in 2003. Following up on her studies, she achieved a PhD in computer science engineering at the Department of Information Technology at the same university. Her research concentrates on the design of multi-sensor architectures, QoS-brokering of novel services, innovative ICT solutions for care, and multi-sensor condition monitoring. Currently, she is an assistant professor at Ghent University and senior researcher at IDLab, Ghent University-imec.